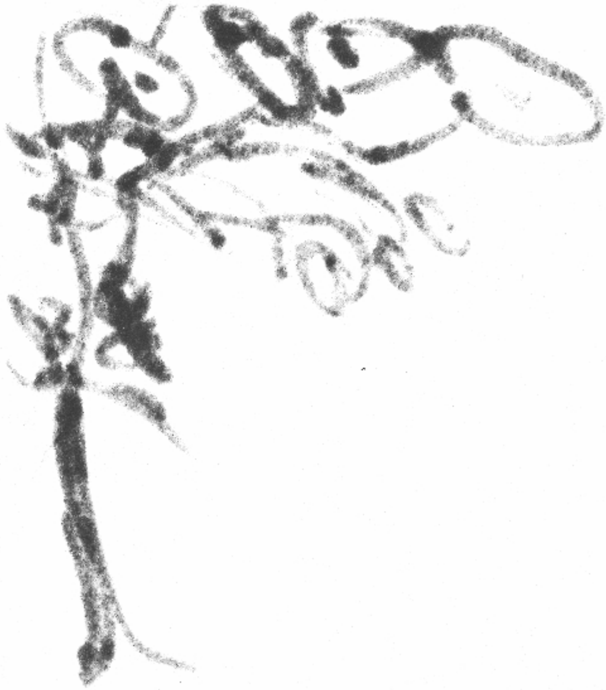
BB Ballantine/31842/$9.95 in USA • $13.50 in Canada

EPSON® PRINTER User's Handbook

The only manual you'll ever need—
from the moment you plug in your printer
through set-up, operation and maintenance

Weber Systems, Inc. Staff

Epson® Printer User's Handbook



Epson® Printer User's Handbook

by
Weber Systems Inc. Staff

Ballantine Books • New York

This book is available to organizations for special use
For further information, direct inquiries to:

Ballantine Books
Special Sales Department
201 East 50th Street
New York, New York 10022

Published in United States by Ballantine Books, a division of Random House, Inc., New York, and simultaneously in Canada by Random House of Canada Limited, Toronto.

Library of Congress Catalog Card Number: 84-90833
ISBN 345-31842-0

Manufactured in the United States of America

First Ballantine Books Edition: February 1985
10 9 8 7 6 5 4 3 2 1

Contents

Introduction

Epson Printer User's Handbook is meant to serve as a tutorial as well as an on-going reference guide to the operation of Epson RX and FX Series printers. Epson's latest innovations are discussed in the text including:

- Character style and density
- Formatting output on the page
- Bit image graphics
- User-defined characters

Also, instructions are included on installing the Epson with a number of widely used personal computers and software packages. A number of examples are included with the text to illustrate the topics being discussed.

Chapter 1 of this book provides an overview of printers in general and an introduction to Epson printers, in particular. Chapter 2 details the steps involved in installing an Epson printer on several popular personal computers. These include: Kaypro II, Sanyo MBC, IBM PC, IBM PC XT, IBM PC*jr*, Commodore 64, Apple IIe, Apple II+, and Compaq. Chapter 3 describes how to install Epson RX and FX Series printers on several popular software applications packages. Covered packages include: WordStar, Lotus 1-2-3, Symphony, dBASE II, pfs:file, and pfs:report.

Chapters 4 through 11 are intended as a tutorial on the use of the codes that control the output of text. An example program is provided for each control code as it is introduced. Chapter 12 contains several examples showing how to accomplish typical output tasks using an Epson printer. Each task is completely covered, from planning through implementation.

Chapter 13 provides an in-depth look at the DIP switches. The DIP switches can be used to customize Epson printers.

Chapter 14 is intended as a tutorial on bit image graphics. Graphics design and specification is covered in detail. Each density is explored, and an example program for each control code is presented. Chapter 15 contains several example programs that use bit image graphics to create larger images. The creation of each graphics image is completely covered, from planning through implementation.

Chapter 16 provides a tutorial on user-defined characters. Character design and specification is covered in detail. Example programs for each code controlling character definition are presented.

Two appendixes are included which detail the Epson ASCII code set and the Epson control codes.

1

Introduction to the Epson Printer

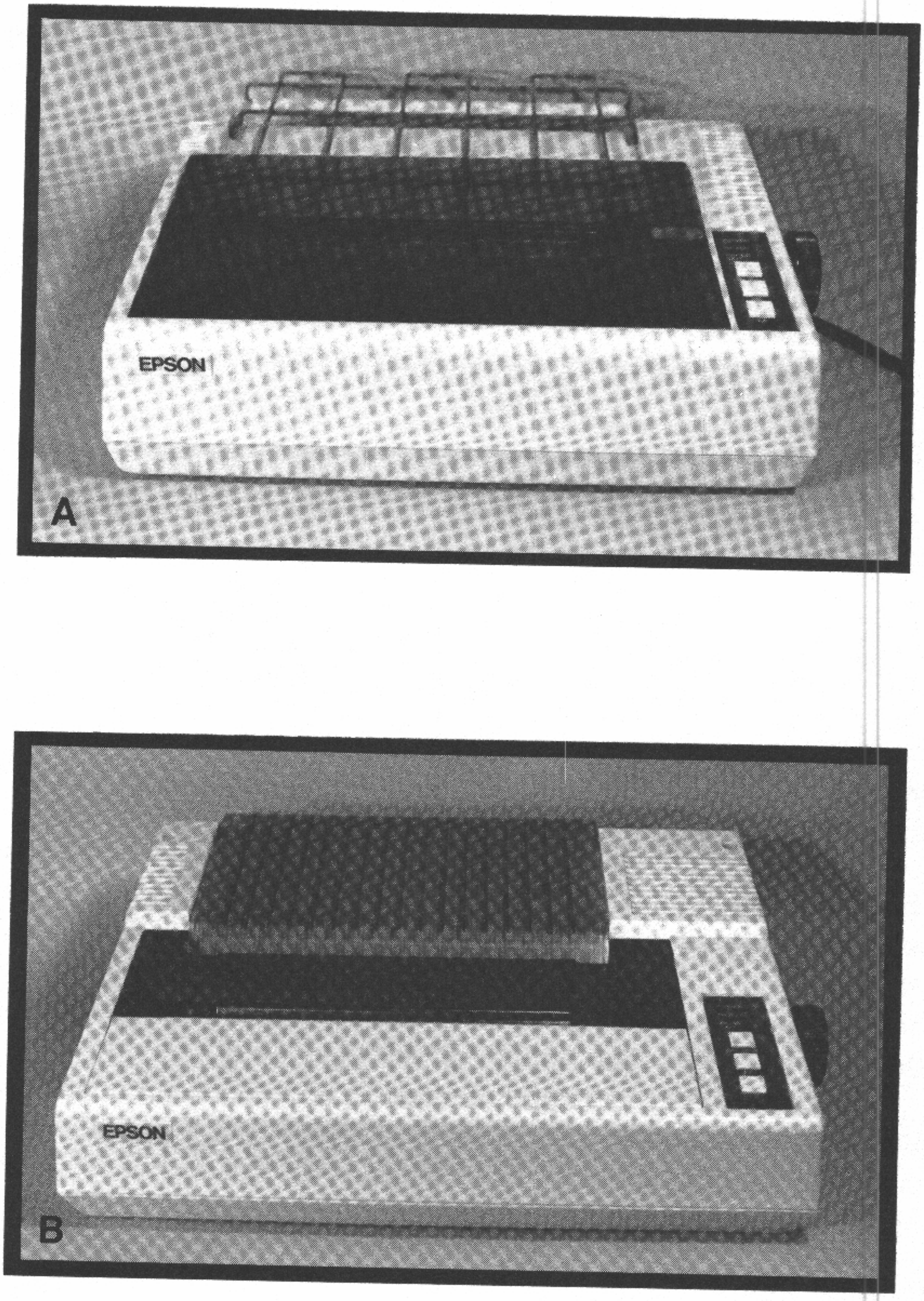
Introduction

This chapter provides all of the information necessary to understand the features of the various Epson printers. The discussion of printer specifications assumes no prior knowledge of printers. The discussion begins with a brief description of the different models of Epson printers. Next, a complete, comprehensive explanation of printers, specifications and other important topics will be presented.

Following that, an in-depth discussion of the Epson printers complete with explanations of controls, print mechanism, etc. will be presented. The chapter concludes with a guide to installation, troubleshooting, and maintenance.

Various Epson Models

There are three series of printers produced by Epson, the MX, RX, and FX. Each model has two versions. In this book, we will discuss only the newer RX and FX models. The FX-80 and RX-80 models are shown in figure 1.1.



a, RX-80; *b*, FX-80

FIGURE 1.1. Epson printers

THE RX SERIES

The RX Series of printers provide an output rate of 100 characters per second.

The difference between the RX-80 and the RX-100 printers is a matter of size. The RX-80 cannot accommodate paper wider than 10 inches, while the RX-100 has a 15½ inch paper width limit.

The RX printers have a total of 10 character sets, plus italics. These printers also have subscript, superscript, and underlining capabilities. The RX printers offer graphics capabilities in three densities.

THE FX SERIES

The FX Series is a line of printers that offer more features than the RX Series. First of all, FX printers output characters at a rate of 160 per second, about twice as fast as their RX counterparts. FX printers also have extended graphics features and additional type styles.

Another outstanding feature of the FX printers is their paper feed system. These printers have both friction feed and tractor feed capabilities. This allows virtually any type of paper to be used with these printers.

Terminology

Printer specifications are often provided in very technical terms. As a result, consumers often find it difficult to evaluate a printer's performance capabilities. This section is intended to explain the various printer "buzzwords" and provide an explanation of the performance of the Epson printers in terms that the lay person can easily understand.

DOT MATRIX

Epson printers do not actually type individual characters. Instead, characters are formed by printing a set of small dots in a specific pattern in order to form a character. Figure 1.2 provides a comparison between typed characters and characters formed by a dot matrix printer.

(a) ABCDEFGHIJKLMNOPQRSTUVWXYZ

(b) ABCDEFGHIJKLMNOPQRSTUVWXYZ

a, Dot matrix characters; *b*, Typed characters

FIGURE 1.2. Dot matrix vs. typewritten characters

Dot matrix printers form characters by printing dots at predetermined locations within a grid. The default character set for Epson printers uses a 9 x 5 matrix for each character. Figure 1.3 depicts some typical printer output with respect to the matrix.

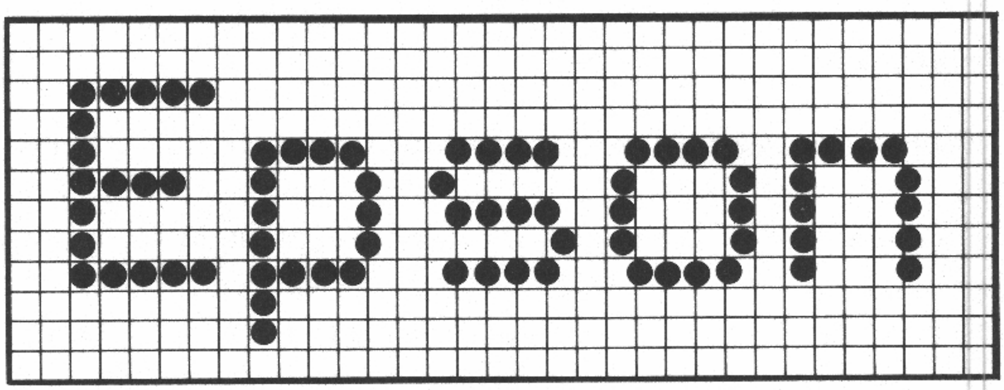


FIGURE 1.3. Printer output superimposed on matrix

Notice that a character can be formed by dots located within each block of the grid, as well as dots located on the vertical lines that separate the blocks. For example, consider the lowercase letter "o". Figure 1.4 details the formation of this character.

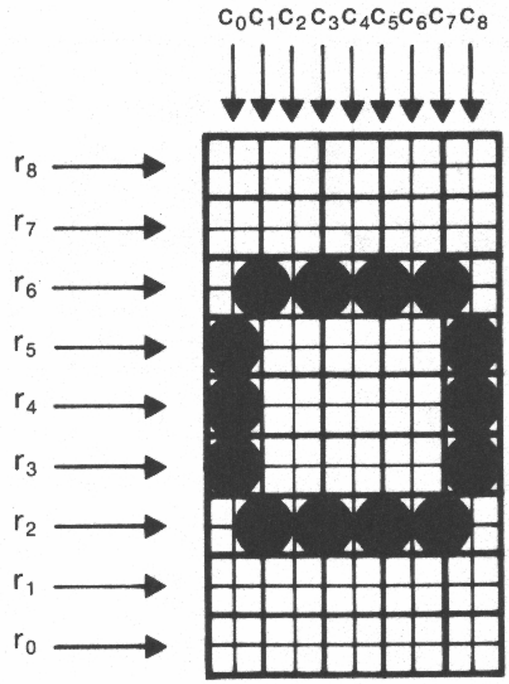


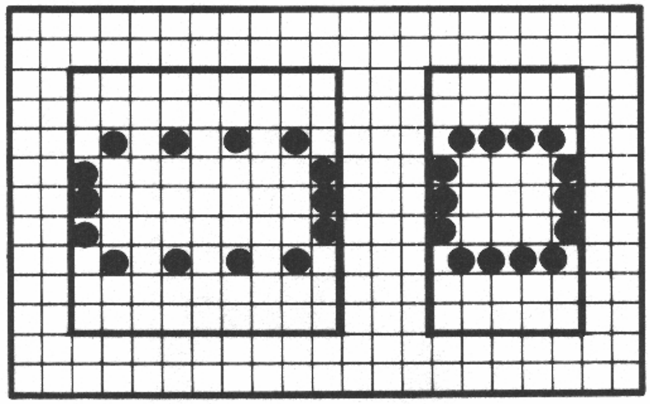
FIGURE 1.4. Character dot positions*

Each dot in a character can be located in any one of the nine rows labeled r_0 through r_8 in figure 1.4. There are also nine possible column locations for each dot. As a result, there are 81 possible dot locations in each character.

An alternative way to visualize the character grid is as a 9 x 9 array, compressed to a size of 9 x 5. Figure 1.5 demonstrates this principle. Note that any dots appearing in adjacent columns will overlap each other in the output.

---

* In computing, counting is usually performed with the natural numbers (0,1,2...) instead of the whole numbers (1,2,3...).



a, 9 x 9 representation; *b*, Compressed 9 x 5 representation

FIGURE 1.5. Dot positions

The formation of a lowercase "o" can be summarized as follows. In column number 0, dots are placed in locations r_3, r_4, and r_5. In column number 1, dots are placed in locations r_2 and r_6. Columns 2, 4, and 6 do not contain any dots while columns 3, 5, and 7 contain the same dots as column 1. The last column has the same dots as column 0.

Dot matrix printers are both economical and versatile. A 9 x 9 matrix allows the printer to display more than 2×10^{24} unique characters. As a result, virtually any type of chart, graph, or illustration can be generated through the proper manipulation of a dot matrix printer. In addition to Epson, many other companies produce dot matrix printers. These include Datasouth, Diablo, NEC, IDS, and Okidata.

LETTER QUALITY

Letter quality is a somewhat elusive term that describes the precision of a printer's output. Epson RX and FX printer output is generally not considered letter quality. However, these models offer a combination of speed, versatility, and cost effectiveness unmatched by any letter quality printer.

The most common type of letter quality printer used with personal computer systems is the **daisy wheel** printer. These types of printers output characters that resemble those output by a typewriter. A round printing element is used that appears much like a spoked wheel. The

wheel spins to the correct position each time a character is to be printed. Daisy wheel printers are generally more expensive than dot matrix printers and output characters at a slower rate. However, the quality of the characters output by daisy wheel printers is higher than that output by dot matrix printers. Diablo, NEC, Brother, as well as many others produce daisy wheel printers.

IMPACT PRINTER

All Epson models are considered impact printers. This term implies that characters are formed by a mechanical device that uses an inked ribbon to transfer an image to the paper. Impact printers allow carbon copies to be produced when an appropriate form is used. Dot matrix, daisy wheel, and IBM ball printers are all considered impact printers, because they are all capable of producing carbon copies. A wide selection of printer forms are available in which several copies can be printed simultaneously. These are commonly known as multipart forms.

Many types of printers are not considered impact printers. For example, thermal printers, ink jet printers, and laser printers are not capable of producing carbon copies.

PRINT HEAD

The print head is the actual device that causes the characters to be printed on the paper. Epson printers feature an inexpensive print head which can be easily replaced when worn. The print head is a precision device that contains a set of 9 dot wires. The dot wires are responsible for the actual placement of dots on the paper.

The dot wires contained in the print head are activated by electrical impulses sent to the print head from the control circuitry. The print head and carriage assembly are depicted in figure 1.6.

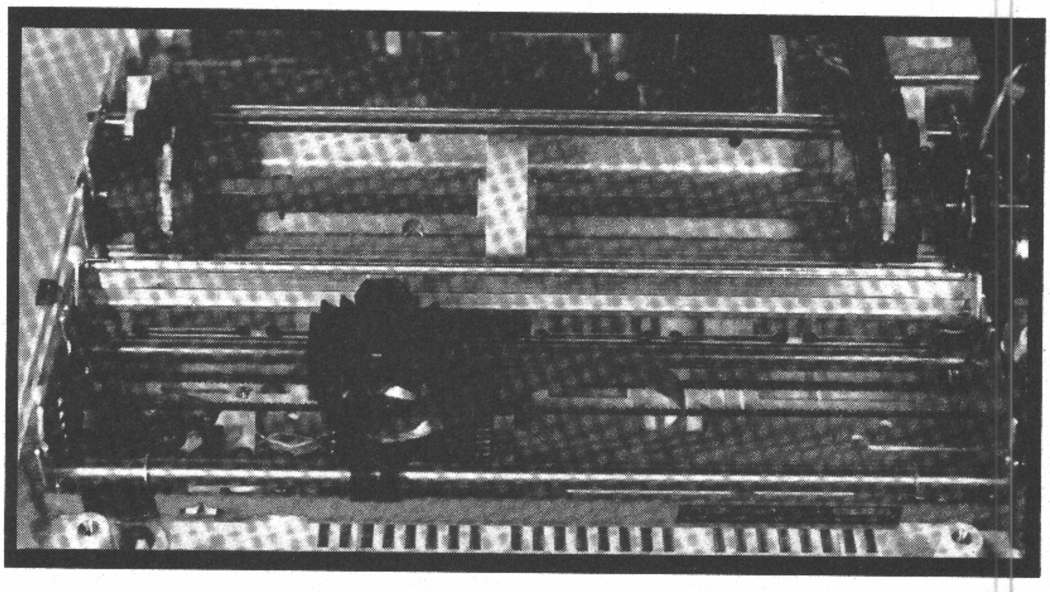
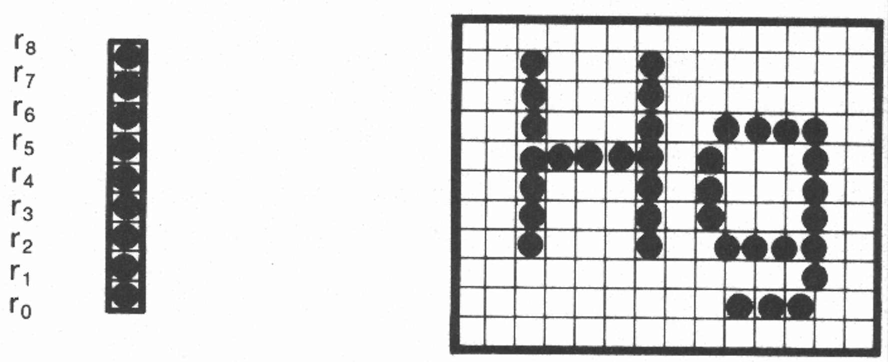


FIGURE 1.6. Print head and carriage assembly

The dot wires are aligned vertically within the print head as depicted in figure 1.7*a*. This configuration allows any character to be formed on the paper if the proper dot wires are activated as the print head moves across the page.

A single dot will be printed on a page whenever a dot wire is thrust against the inked ribbon. Characters as well as graphics displays can be generated through the control of the dot wires. For example, consider the output depicted in figure 1.7*b*.



(a) (b)

FIGURE 1.7. *a,* Dot wire arrangement; *b,* Typical output

BIDIRECTIONAL PRINTING

The carriage is the assembly within the printer that contains the print head. The motion of the carriage is responsible for the placement of the characters on the page. Epson printers allow characters to be output while the carriage moves in either direction. This feature is known as **bidirectional printing.**

Bidirectional printing enhances the output rate of the printer since the carriage does not need to return to the left margin before printing a line. Instead, the printer alternates between printing from left to right and right to left.

FRICTION FEED

A friction feed system is the method used to advance paper through the printer. This type of paper feed is the same as that used in standard typewriters. The paper being sent through the printer is pinched between the platen and another roller. Whenever the platen is rotated, the printer paper will be advanced.

Friction feed systems allow the use of virtually any type of paper or form that can be used in an ordinary typewriter. Friction feed systems do not require the use of paper with holes along the edges.

PIN FEED OR TRACTOR FEED PAPER

Pin feed or tractor feed paper refers to printer paper that has holes along both sides. Many printers have sprockets to control the advance of the paper through the printer. As a result, pin feed paper has become a standard item not only with various paper sizes, but with envelopes, invoices, checks, mailing labels, and order forms as well.

Many computer equipment vendors offer customized invoices, checks and other forms. Business or personal addresses can be printed on mailing labels, envelopes, etc. Many forms are also offered in a multipart format that allows several copies to be printed simultaneously. A typical collection of forms available in pin feed format is pictured in figure 1.8.

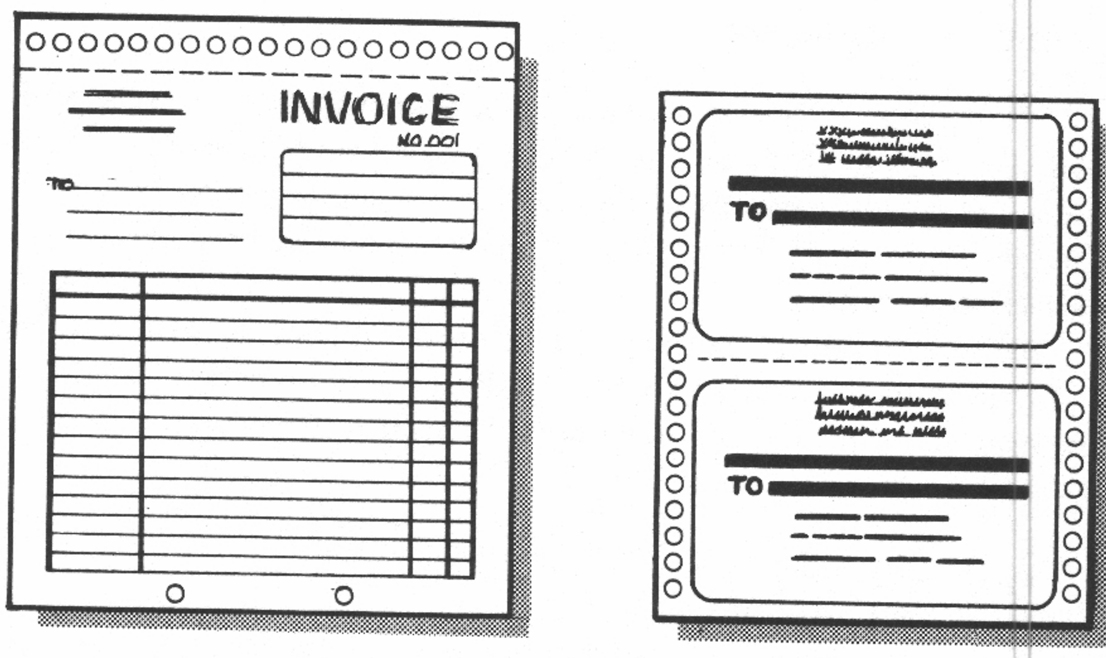


FIGURE 1.8. Common printer forms

FANFOLD PAPER *

Fanfold paper is widely used with computer printers. Fanfold paper is one continuous sheet which has been divided into individual pages by perforating the sheets (see figure 1.9). This allows the paper to be fed continuously through the printer, yet also allows individual pages to be easily separated by tearing the perforations.

The job of separating fanfold paper or multi-part forms can be tedious and time-consuming if attempted by hand. Fortunately, machines known as **bursters** and **decollators** are available for separating continuous forms into separate sheets. A typical burster is depicted in figure 1.10.

* The fanfold concept is applicable to forms as well as paper. These are generally referred to as continuous forms.

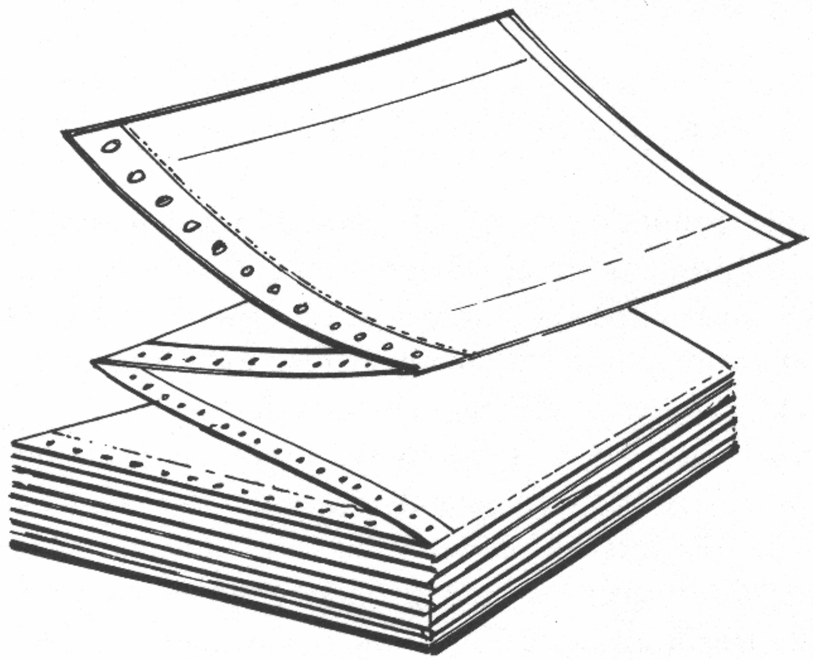


FIGURE 1.9. Fanfold paper

FIGURE 1.10. Burster

PARALLEL INTERFACE

Parallel interface is a term that describes a type of connection between a printer and a computer. All Epson printers have a standard parallel interface built in.

Parallel communications is a method in which all the information necessary to produce one character (1 byte) is sent to the printer simultaneously. Since one byte of information consists of 8 bits (or digits), a parallel interface requires 8 individual data lines. A parallel communication system is depicted schematically in figure 1.11.

Parallel interface connections are widely used in small computer systems where system components are only separated by a few feet. Standard parallel interface connections commonly use a type of connector called "Centronics." Epson printers contain a female Centronics connector, which allows an easy hookup to many computer systems. Figure 1.12 depicts a Centronics connector.

The cable used to connect the printer to a computer is not included with the printer. Printer cables will be discussed in detail in chapter 2.

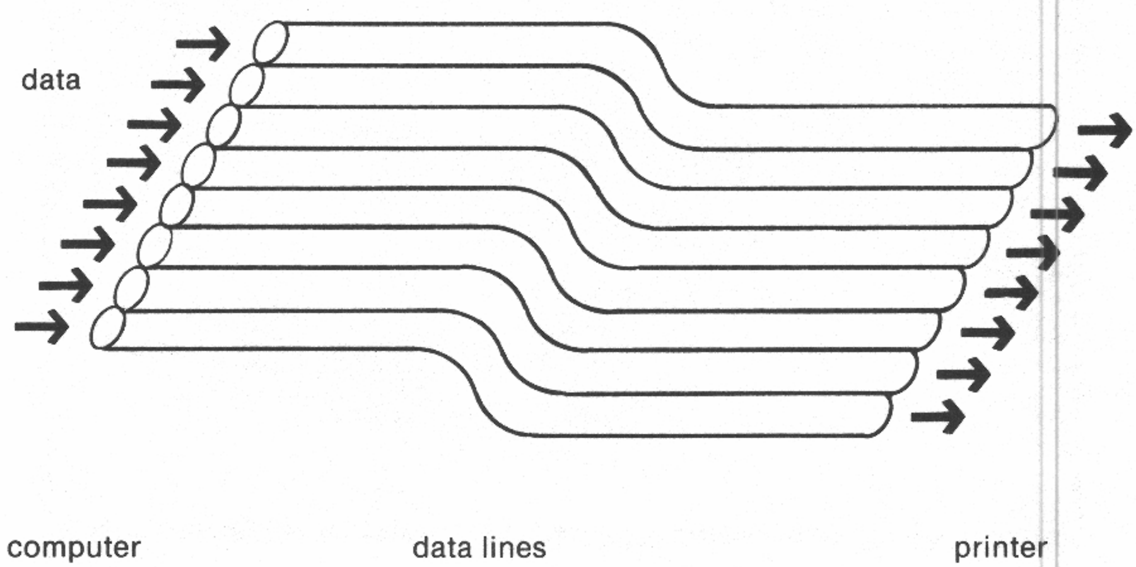


FIGURE 1.11. Parallel data transmission

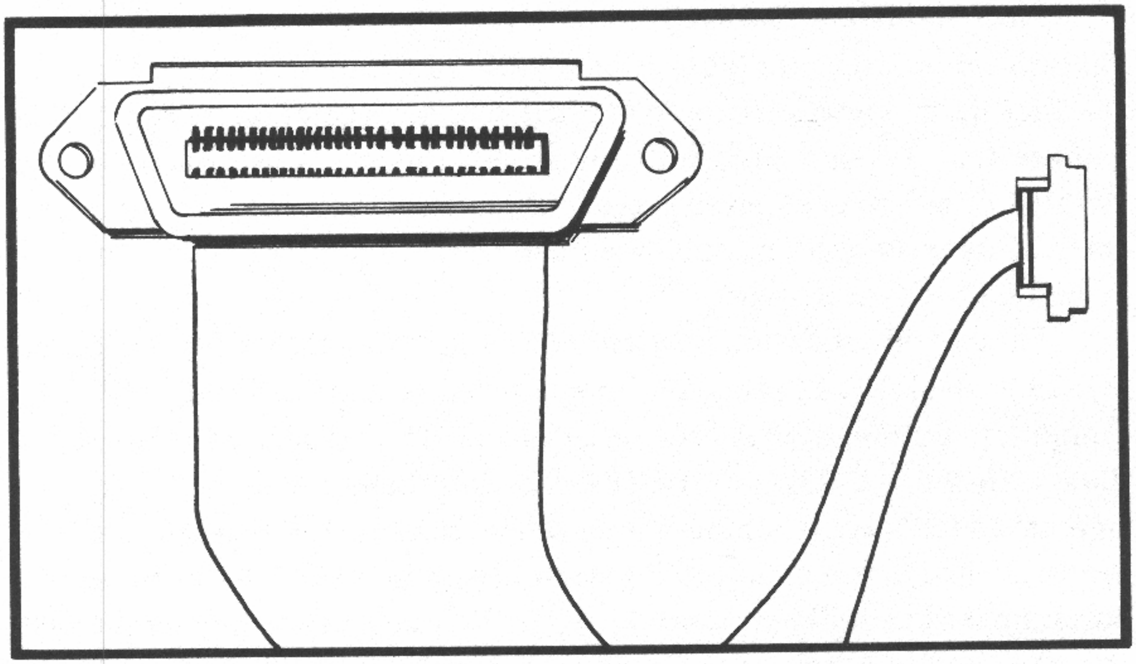


FIGURE 1.12. Centronics connectors

SERIAL INTERFACE

Serial interfacing is a technique that can be used to connect a printer to a computer system. Serial data transmission is a method in which each bit of data is sent individually along a single data line. Recall that 8 bits of data are required to define a single character. Serial data transmission is represented schematically in figure 1.13.

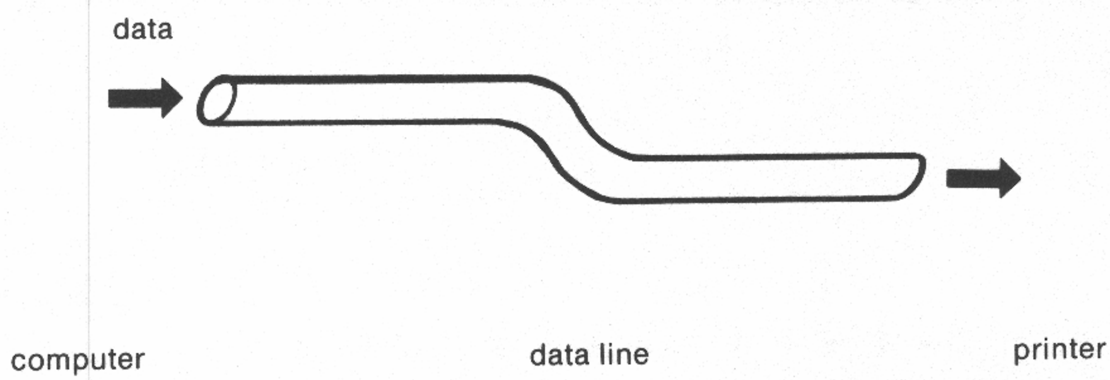


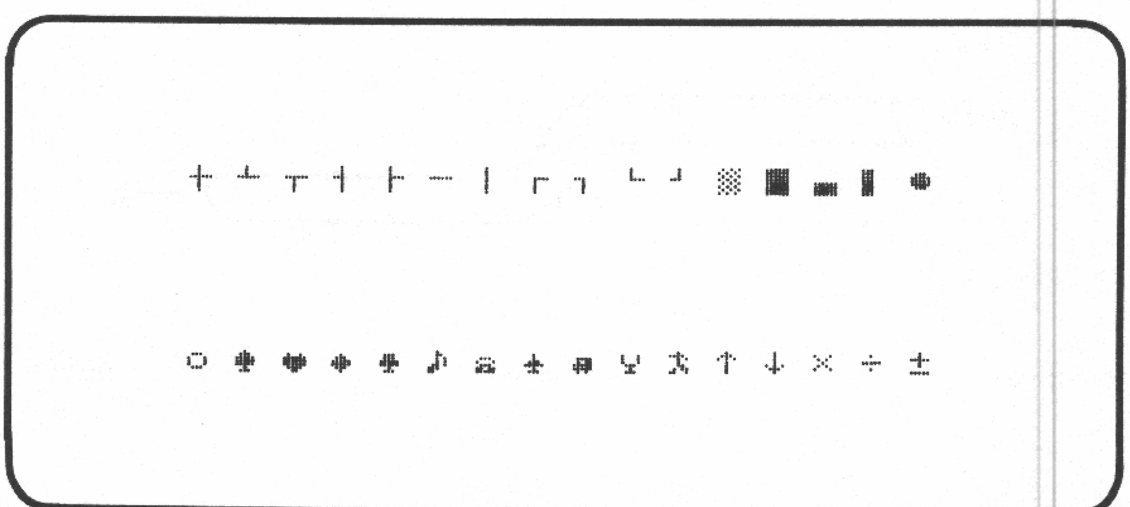
FIGURE 1.13. Serial data transmission

Since serial interfaces require each bit to be sent individually (rather than as sets of 8), serial data is generally transferred more slowly than parallel data. However, the electrical characteristics of parallel data require that all data lines be kept as short as possible. Rarely will a parallel interface cable ever exceed 12 feet in length. As a result, serial interfacing techniques must be used whenever computer components are separated by large distances.

The two most commonly used serial interfacing techniques are called RS-232C and IEEE 488. All Epson printers can be interfaced with computer equipment that uses either of the aforementioned formats for data transfer. RS-232C and IEEE 488 interfacing equipment can be purchased from most computer equipment dealers, but these devices are not furnished with any of the standard Epson models. If your computer equipment does not have a standard parallel output for a printer, be sure that the proper interfacing equipment is available before purchasing an Epson printer.

GRAPHICS

Some printers are not restricted to the production of letters, numbers, and special characters. Epson printers, for example, can be used to generate charts, graphs, illustrations, etc. A printer's ability to generate various forms of output is known as **graphics.** All Epson printers have excellent graphics capabilities. Examples of the graphics character set are given below:



BUFFER

Because it is basically a mechanical device, the printer handles data much more slowly than the computer itself or the other peripherals in a computer system (i.e. disk drives, video display, modem, etc.). If the computer had to wait until data transferred to the printer had actually been output, the entire system would in effect be paralyzed.

To circumvent this problem, **buffers** * are incorporated into printers. In the context of a printer, a buffer can be described as self contained memory to which data can be transferred from the computer's memory. Information is held in the printer's buffer until it has been processed.

Specifications

The preceding explanation of printer terms allows the specifications of the Epson printers to be presented here in a somewhat more scientific manner.

All models of the Epson printers have the following features:

- Dot matrix (impact, output)
- Replaceable print head
- 9 independent dot wires
- Bidirectional printing (in text mode)
- Logic seeking print head
- Standard parallel interface
- Centronics connector

The RX Series has the following features:

- 10 character sets
- Output rate of 100 cps
- 6 type sizes
- Single, double, and quad density graphics
- Boldface, italics, underline, subscripts, and superscripts

---

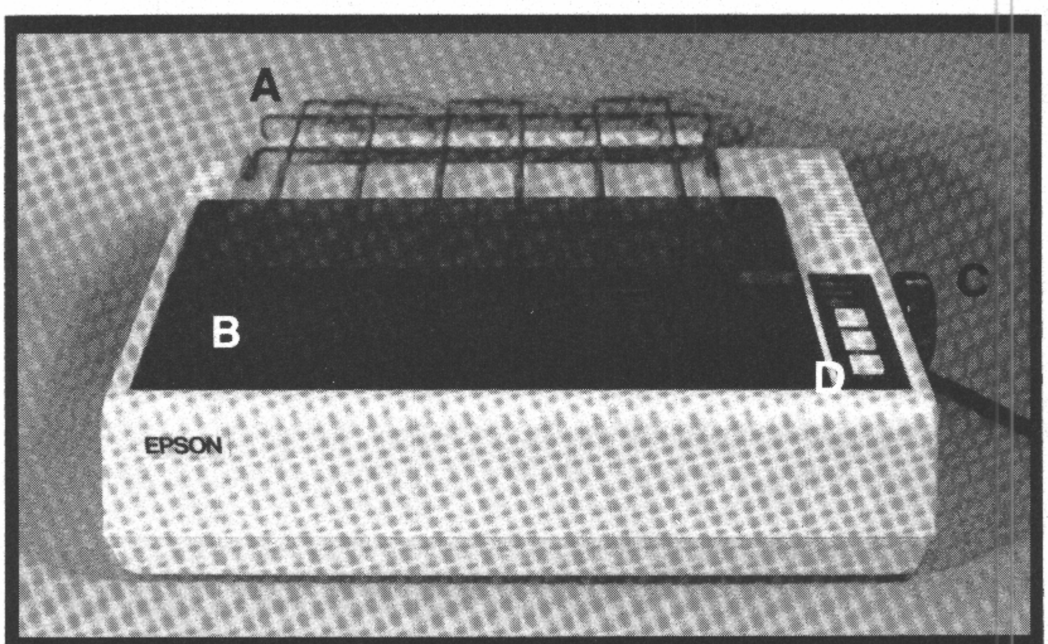
* Generally, the computer itself will reserve an area in its internal memory where data to be sent to the printer is stored. This data is then transferred in blocks from the computer's memory to the printer buffer.

The FX Series has the following features:

- 10 character sets
- Output rate of 160 cps
- 6 type sizes
- Extended buffer
- Single, double, and quad density graphics
- Boldface, italics, underline, subscripts, and superscripts

Anatomy of a Printer

A typical Epson printer is pictured in figure 1.14. This top view points out some of the important parts of the printer.

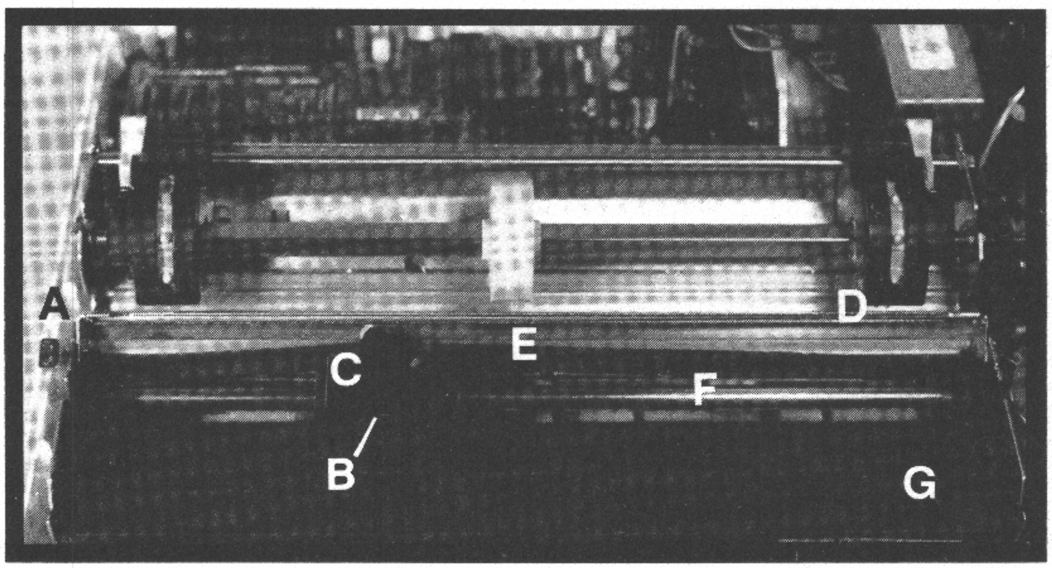


a, Paper separator; *b,* Cover; *c,* Manual advance knob; *d,* Switches and indicators

FIGURE 1.14. Epson printer (RX-80)

The power switch is located on the rear portion of the right side of the printer. The paper advance knob is also located on the right side. The ON LINE, LF (line feed), and FF (form feed) switches are located on the top of the printer in the front right corner. This panel also contains a set of indicators for power, on line, paper, etc. The cover on the Epson printer performs three main functions. First of all, the cover prevents debris from damaging the precision parts of the print mechanism. Next, the cover reduces the amount of noise generated by the printer. Lastly, the cover provides an edge that allows paper to be neatly torn off. The paper separator is a device that allows a clear paper path for the incoming and outgoing paper.

The print mechanisms for all Epson printers are similar. The diagram in figure 1.15 shows the locations of many important components.



a, Impression control lever; *b*, Carriage; *c*, Print head; *d*, Pressure bar; *e*, Ribbon; *f*, Carriage track; *g*, Ribbon cartridge

FIGURE 1.15. A typical Epson print mechanism

The pressure bar is a spring loaded mechanism that keeps the paper pressed firmly against the printing surface. The pressure bar ensures more accurate, better looking output. The ribbon is an inked material that allows an impact from the print head to cause an output on the paper. The carriage track is a precision, tubular rail that guides the motion of the carriage. The ribbon cartridge is a self contained package that allows the printer ribbon to be changed quickly and easily.

The print head is the device that causes the characters to be formed on the page. As mentioned earlier, the print head contains nine dot wires. When the dot wires become worn, the print head can be easily replaced. This feature allows an Epson printer to produce quality output regardless of its age or usage.

The print head is mounted on the carriage. The carriage moves from side to side within the printer while output is being produced.

The impression control lever is used to adjust the darkness of the printer's output. When the lever is moved toward the front of the printer, the output will become darker. Similarly, the output becomes lighter when the lever is moved toward the back of the printer.

CONTROLS

Five external controls are available for manipulating the functions of the various Epson printers. The five controls are as follows:

- Power switch
- ON LINE switch
- LF switch
- FF switch
- Paper advance knob

Power Switch

The power switch is used to activate the printer.

ON LINE Switch

The ON LINE switch is used to switch the control of the printer from the computer to the controls on the printer itself. When the printer is ON LINE, it lies fully under the control of the computer. When the computer is not on line, the printer can only be controlled by its control panel. Each time the on line switch is pressed, the printer will change from on line to off line, or vice versa.

LF Switch

The LF switch causes the paper in the printer to be advanced by one line. Holding down the LF switch causes the paper to be advanced repeatedly. The LF switch has no effect when the printer is on line.

FF Switch

The FF switch causes the paper in the printer to be advanced to the top of the next page. Like the LF switch, the FF switch can only be used when the printer is not on line.

Paper Advance Knob

The paper advance knob allows the paper in the printer to be advanced manually. The paper should be advanced in this manner when the printer is not on line or not powered on. Never force the paper advance knob. If the knob locks in place, turn the printer off before turning the knob.

Installation

Epson printers are very easy to install. Each printer is supplied with a manual that contains a complete description of the installation procedure. The installation procedure consists of five steps. The first step is to remove the printer from its packing materials. The second step is to remove the screws used to secure the internal components of the printer during shipping. The third step consists of the installation of a few additional parts. The fourth step consists of placing the printer at a convenient location and connecting it to an outlet and a computer system. The fifth step consists of the insertion of an appropriate type of paper.

When the printer is received, inspect the carton immediately for evidence of damage during shipping. If no damage is apparent, carefully remove the printer from its carton and place it on a firm surface where the packing material can be removed.

Next, remove the screws used to secure the internal mechanism. Consult your printer manual for the exact location of these screws. Remove the black paper advance knob from its packaging and press it onto the paper advance shaft on the side of the printer. Note that the knob can only be installed in one position since the shaft has a flattened edge. Be sure to observe the position of the knob during installation. Otherwise, the knob might be broken.

Epson printers are also supplied with a paper separator. This device prevents the outgoing paper from being obstructed by the incoming

paper. The details of the installation of the paper separator are contained in the printer manual.

The installation of a ribbon cartridge is the final step in the assembly of the printer. Remove the printer cover by rotating it to its vertical position. Then, gently lift it out of its hinges. Move the pressure bar against the paper path so it does not obscure the print head. Remove the ribbon cartridge from its packaging and place it in its proper location (see figure 1.15). Gently press down on the cartridge until it snaps into position. Be sure that the ribbon is located in front of the print head. Consult your printer manual for a more detailed guide to the installation of ribbon cartridges.

Next, place the printer on a firm surface near your computer system. Like all computer equipment, locate the printer where it will not be exposed to extremes of temperature or high humidity. Also, protect the equipment from dust, dirt, and other hazards. Locate the printer near a standard electrical outlet and insert the printer's power cord plug in the outlet. If the computer has a cable with a Centronics plug for printer output, simply insert the plug in the receptacle in the back of the printer. In most cases, an appropriate printer cable will have to be purchased separately. The exact type of cable required will depend upon the type of computer system. Consult a computer equipment dealer for details concerning printer cables. Keep in mind that Epson printers are designed to operate with a standard parallel interface. If your computer has some other type of output, you will need additional interfacing equipment.

Once the printer has been properly connected, paper can be inserted, and the printer can be operated. Consult your operations manual for the exact information concerning the loading of paper.

Power On Sequence

Once the printer has been properly installed and contains paper, it will be ready to be powered on. As mentioned earlier, the printer's FF switch causes the paper to advance to the top of the next page. In order for this feature to work properly, use the paper advance knob to set the top of the page at the location of the print head. If this procedure is followed, the FF switch will always have the proper effect. The form feed feature operates under the assumption that the paper in use is 11 inches long.

When the top of the page has been properly set, the printer will be ready to be powered on. This can be accomplished by simply moving the power switch to the ON position.

A test procedure can be used to verify that the printer is operating properly. This test procedure can be performed by simply holding down the LF switch while turning the printer's power on. This procedure results in the output of the entire standard character set. The diagnostic procedure is automatically repeated, so the output of the test procedure would appear as depicted in figure 1.16.

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJK
$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKL
%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLM
&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN
```

FIGURE 1.16. Diagnostic procedure output

The diagnostic procedure can be stopped by turning the printer off. If the results of the diagnostics are acceptable, the printer will be ready for use. If the results of the test are unsatisfactory, check that the ribbon cartridge has been installed properly. If the problem persists, consult your computer equipment dealer or factory authorized service personnel.

Maintenance

Epson printers generally require very little maintenance. The only three procedures described here are cleaning, replacing a ribbon cartridge, and replacing a print head. Any other problems with a printer should be referred to factory authorized service personnel.

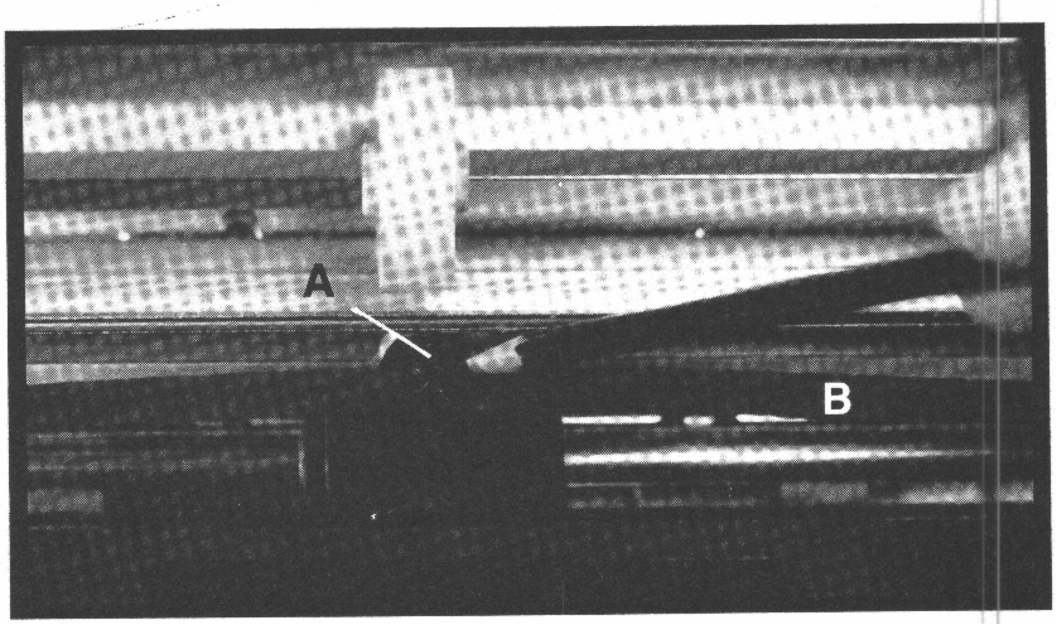
CLEANING

The manufacturer recommends a cleaning of the printer after every three months of operation. It is not necessary to disassemble the printer while cleaning. Instead, use a clean, soft brush to clear dust and debris from the printer mechanism. The exterior surfaces of the printer can be cleaned with a combination of water and mild detergent. Use the cleaning solution sparingly so that no water will leak into the printer. Such leakage could cause a great deal of damage.

RIBBON CARTRIDGES

Although ribbon cartridges are usually good for two million characters, they eventually will need to be replaced. Since the cartridges merely snap into place, a ribbon cartridge can be removed by pulling the cartridge up from its mounting within the printer. Be sure to move the pressure bar against the paper before attempting to remove the cartridge.

The cartridge can be replaced by simply placing it in the appropriate location and pressing down until it snaps into place. When installing a ribbon cartridge, be sure that the ribbon is placed in front of the nose of the print head. A correct installation is detailed in figure 1.17.



a, Print head nose *b,* Ribbon

FIGURE 1.17. Printer ribbon installation

PRINT HEAD

Epson printers feature a replaceable print head that allows them to perform like new, regardless of age. If one of the dot wires becomes worn, the print head should be replaced.

To remove a print head, remove the ribbon cartridge and locate the head lock lever (see figure 1.18). Rotate the lever clockwise approximately one quarter turn, or 90°.

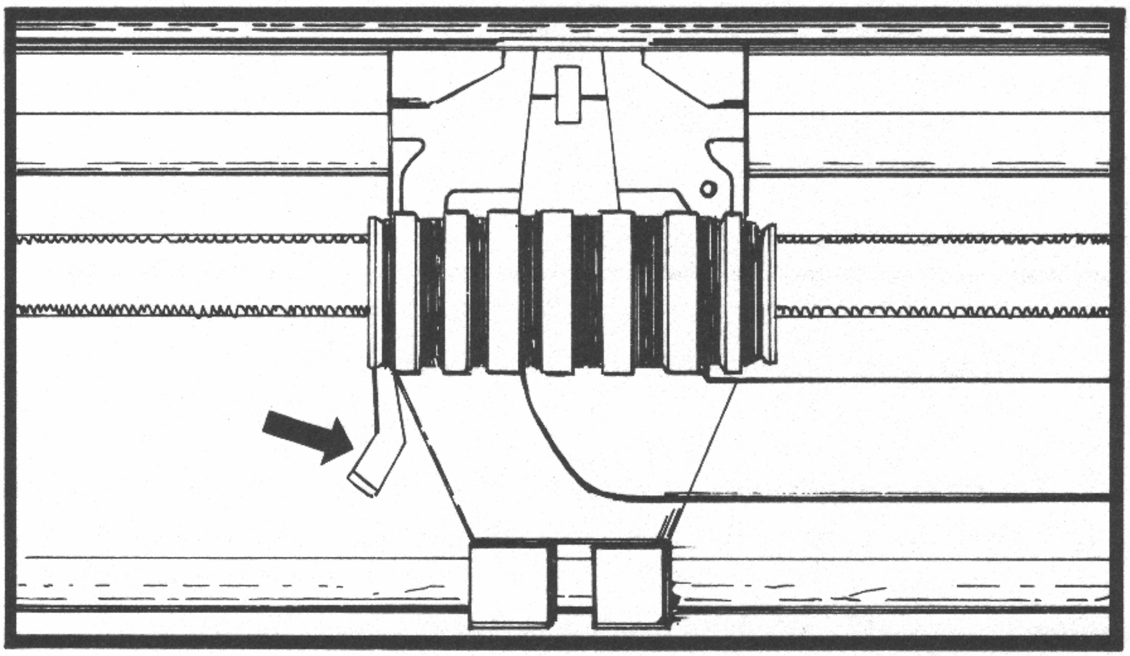


FIGURE 1.18. Head lock lever (top view)

Once the lock lever has been released, the print head can be removed by pulling upward on the print head unit. Do not move the carriage assembly when the print head is not in place.

The print head is attached to a flat "ribbon" cable. As a result, this cable must be disconnected before the print head can be completely removed. The cable cannot be removed from the print head, so it must be disconnected from the printer instead. The connector for the print head cable is located in the center of the printer. This connector pinches the exposed conductors at the end of the cable with terminals in the connector (see figure 1.19).

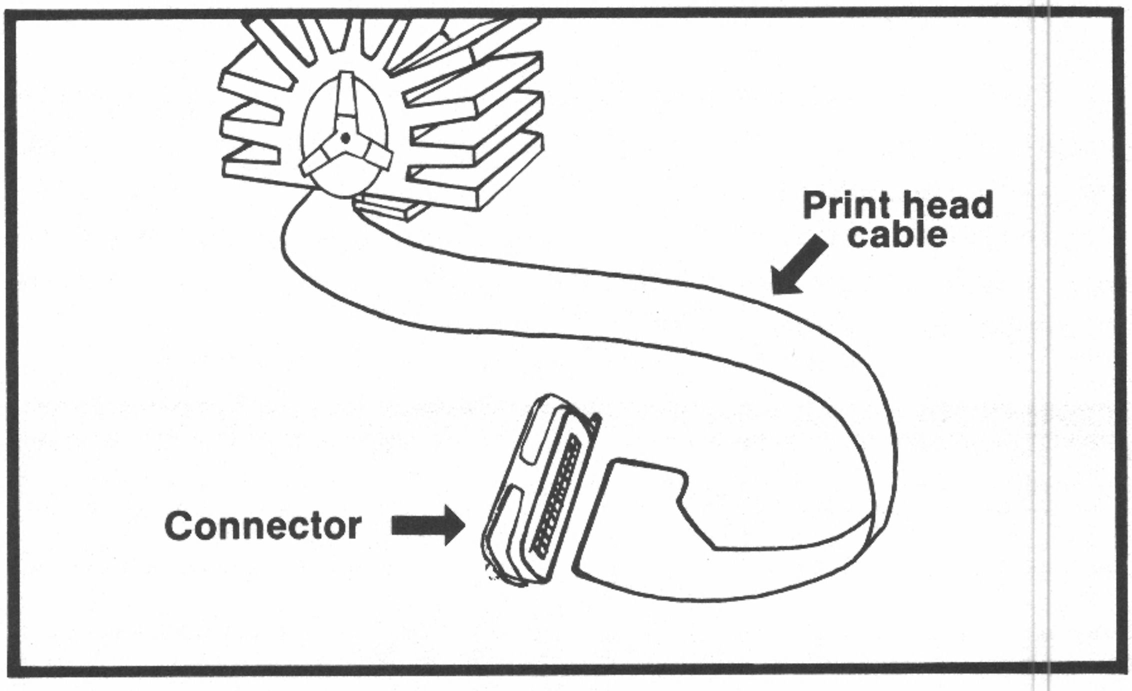


FIGURE 1.19. Print head cable and connector

The ribbon cable has a sturdy tab attached to its underside. This tab allows the cable to be easily pulled from the connector. Grab the tab, and pull it firmly toward the right. The ribbon should then be freed from the connector. Install a new print head by performing the above procedure in reverse.

Troubleshooting Guide

The following table contains a list of some commonly encountered problems along with some possible causes and solutions.

Table 1.1. Troubleshooting guide

Problem	Possible Cause	Solution
No power	Printer not plugged in	Plug in printer.
	Outlet controlled by wall switch	Turn wall switch on or move printer to another outlet.
Badly formed characters	Ribbon cartridge improperly installed	Be sure that ribbon cartridge is locked in place.
	Worn print head	Replace print head.
	Printer cable not connected properly	Be sure that both ends of printer cable are securely plugged in.
No output	Disconnected cable	Make appropriate connection.
	Programming error	Check computer manuals for techniques used to send data to printer.
	Printer not on line	Use ON LINE switch to illuminate the ON LINE indicator.
	Printer out of paper	Insert appropriate printer paper.
	Printer not compatible	Consult a computer equipment dealer for an appropriate cable or interface.

2

Installing Epson Printers on Personal Computers

Introduction

In this chapter, we will discuss the steps involved in installing the Epson printer on several popular models of personal computers. Information about DIP switch settings, interfacing material, and cables will be included in this discussion. We will also include step-by-step instructions on how to make the physical connections and a test program that can be used to check the computer/printer interface.

DIP Switches

Epson printers, when used with any of the computers discussed in this chapter, will correctly generate output using the factory settings of the DIP switches. Figure 2.1 illustrates these settings for both the RX and the FX Series. For more information about DIP switches, see chapter 13.

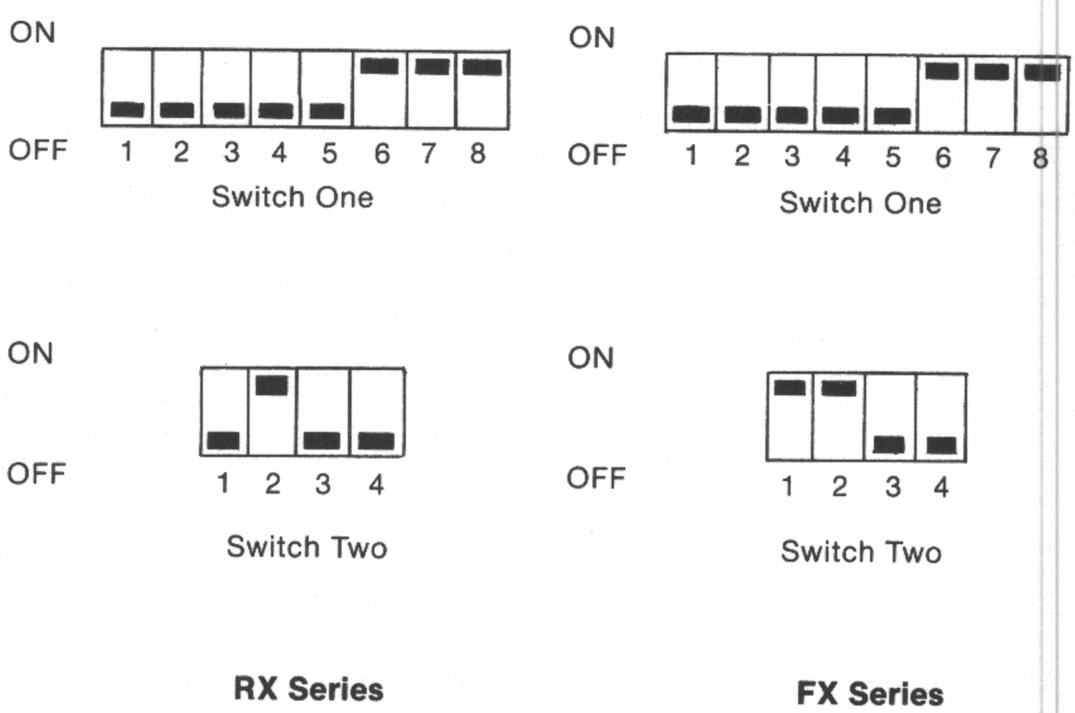
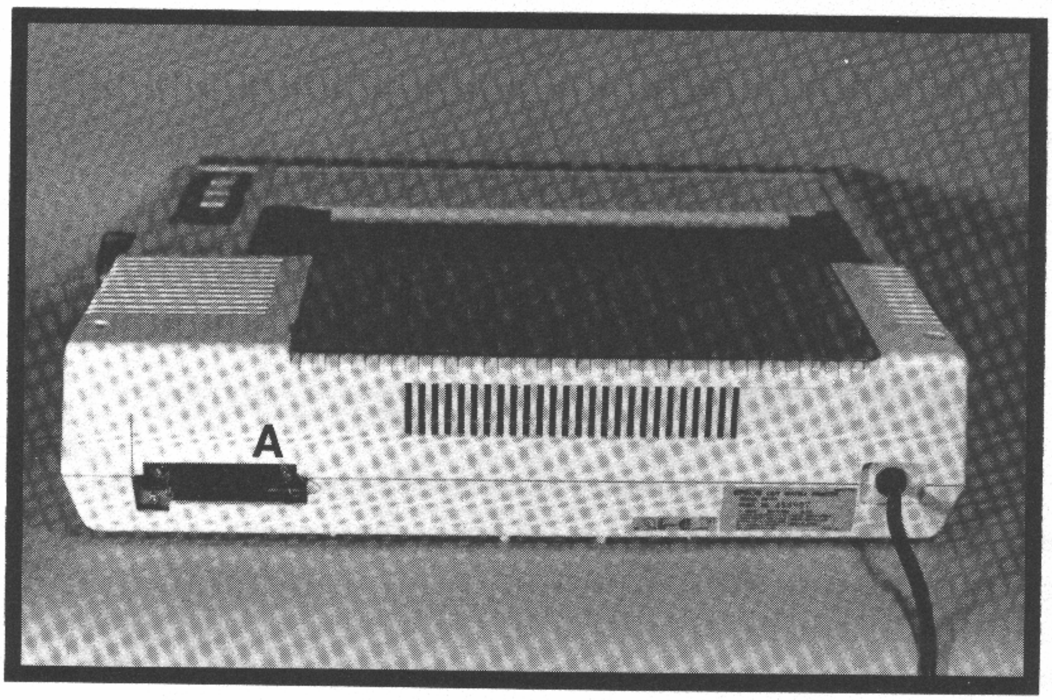


FIGURE 2.1. DIP switch settings for the RX and FX Series

Connecting the Cable to the Printer

The section on each specific computer details the type of cable needed to accomplish hooking up the printer. All of these cables have at least one end with a Centronics connector attached. After connecting the cable to the computer, attach this Centronics connector to the parallel port in the back of the printer. Only one alignment can be used to make the connection. The wider side of the connector must be facing up. Figure 2.2 identifies the position of the port in the back of the printer.



a, Parallel port

FIGURE 2.2. Locating the port in the back of an Epson printer

Apple IIe and II+

The addition of an Apple parallel interface card enables the Apple IIe® and Apple II+® computers to drive an Epson printer. Figure 2.3 contains a picture of the parallel interface card. This card is installed by inserting it into one of the available expansion slots. For the purposes of this example, we installed the card in slot #1.

CABLE

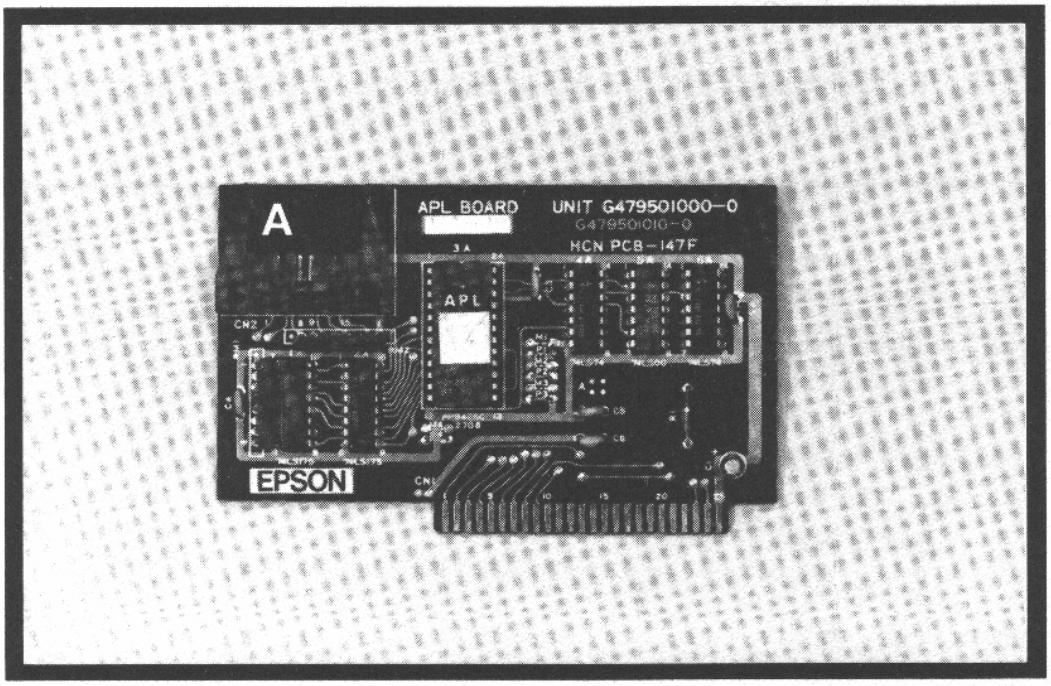
To accomplish the installation of the printer, a cable is also required. The required cable is a ribbon cable with a male Centronics connector attached to one end and a small 15 pin cable attached to the other. This cable is known as an Apple cable. Both of these connectors are asymmetrical. This fact forces the connection to be made correctly. Figure 2.4 contains a picture of the cable.

HOOK UP

Verify that both the printer and the computer are shut off before hooking the printer to the computer.

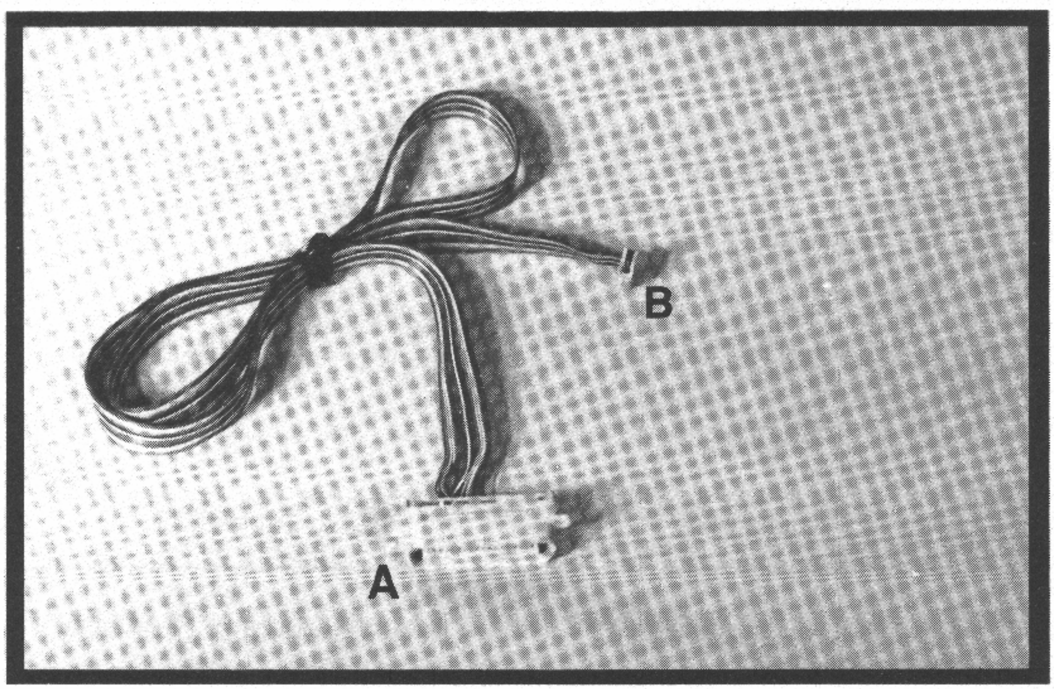
First, connect the end of the cable attached to the small connector to the parallel interface card. Figure 2.3 on the opposite page illustrates the location on the card where the connection must be made. Note that only one alignment can be used to make the connection. The side of the connector from which the cable extends must be facing away from the board.

Now connect the other end of the cable to the port in the back of the printer, as discussed at the beginning of this chapter.



a, Cable connection

FIGURE 2.3. Apple parallel interface card



a, Centronics connector; *b,* Small connector

FIGURE 2.4. An Apple cable

TEST PROGRAM

The following BASIC program can be used to test the computer/printer interface:

```
10   PR# 1
20   FOR I = 1 TO 11
30   READ SPEC
40   PRINT  CHR$ (27) CHR$ (SPEC);

50   PRINT "TESTING 1,2,3 TESTING"

60   NEXT I
70   PR# 0
80   END
90   DATA  77,80,14,52,15,69,53
100   DATA  49,50,70,64
```

The program output is:

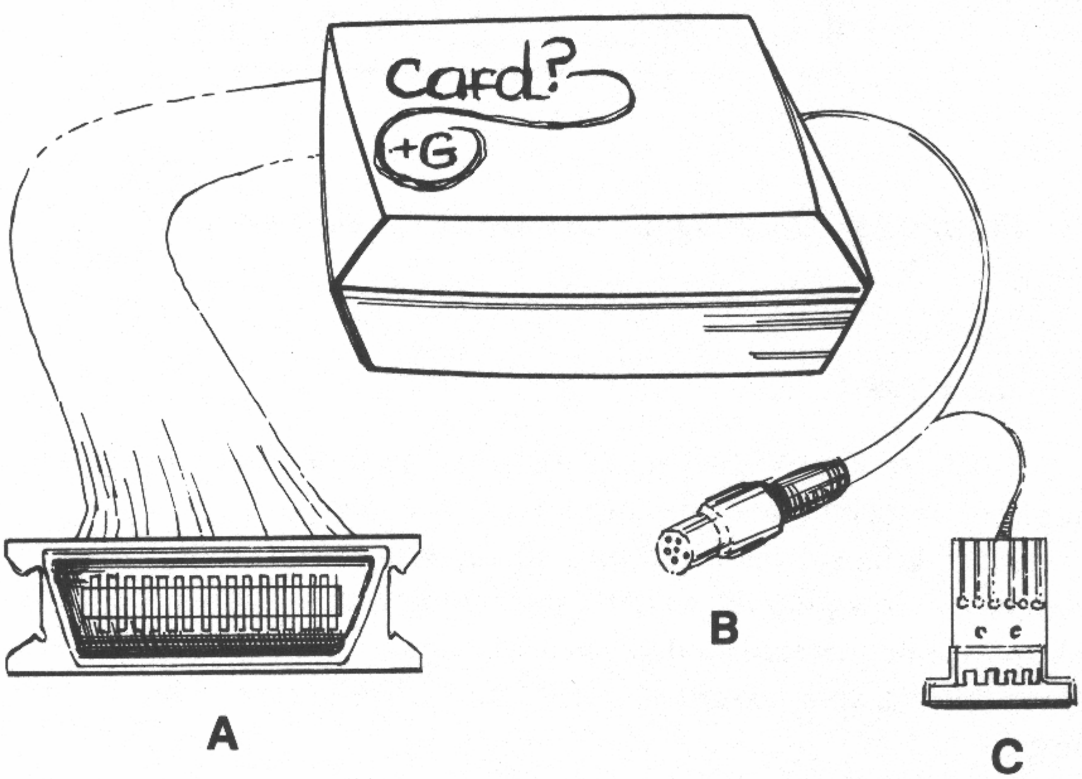
```
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
T E S T I N G   1 , 2 , 3   T E S T I N G
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
```

For now, do not worry about how the program functions. Just verify that the output you obtain is identical to the output shown above.

Commodore 64

The Commodore 64® computer requires a special interface to drive an Epson printer. We used a Card ?/+G manufactured by Cardco, Inc. This interface includes all the cables necessary to hook an Epson printer up to a Commodore 64. Figure 2.5 contains a picture of the card.

Before connecting the interface, set the DIP switch in the interface to its proper configuration. To locate the DIP switch, set the interface on its top. The top is the side with the printing on it. Remove the four screws from the case and lift the bottom case off of the rest of the interface. Figure 2.6 shows the location of the DIP switch inside the interface. This figure also shows the proper setting of the interface's DIP switch for use with an Epson FX or RX printer. The DIP switch must be set because this interface was designed to operate with several different printers.



a, Centronics connector; *b*, Six pin plug; *c*, Cassette port connector

FIGURE 2.5. Card ?/+G manufactured by Cardco, Inc. to interface Epson printers with Commodore 64 computers

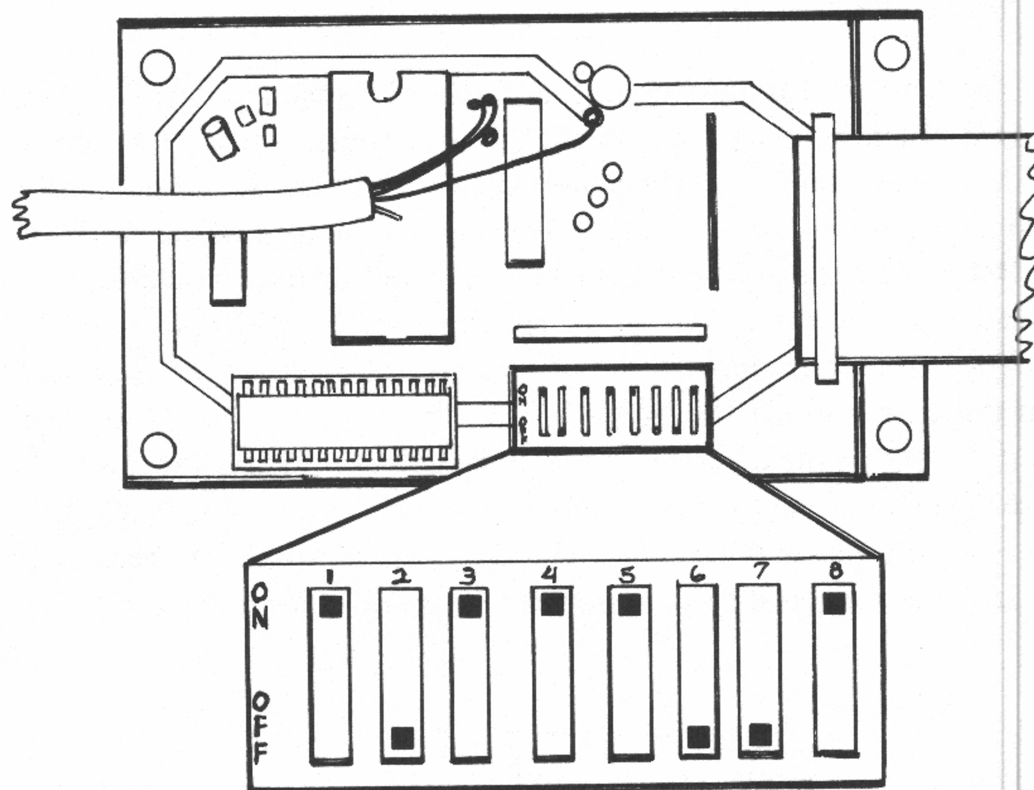


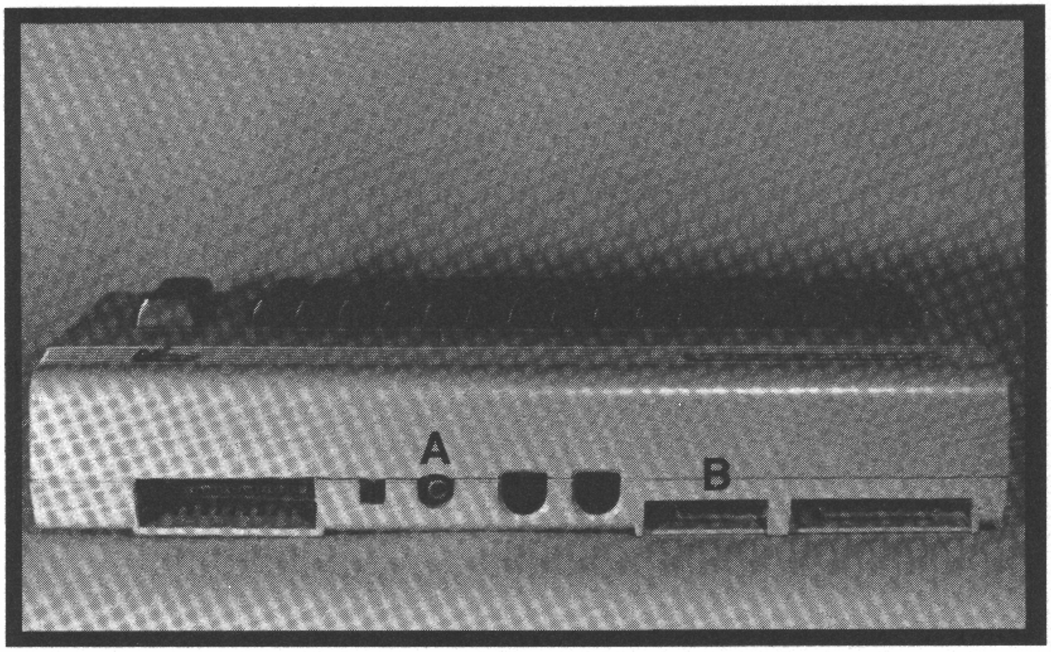
FIGURE 2.6. Locating the DIP switch inside the interface

HOOK UP

Verify that both the printer and the computer are shut off before hooking the printer and the computer to the interface.

First, connect the 6 pin plug to the 6 pin port in the back of the computer. Now plug the cassette port connector into the cassette port. The plastic connector should be below the board. The board may be used to reconnect a data cassette player. Figure 2.7 shows the location of these ports.

Now connect the cable with the Centronics connector attached to it to the port in the back of the printer, as discussed at the beginning of this chapter. The manufacturer states that the printer must always be powered on before the computer when using this interface.



a, Six pin port; *b*, Cassette port

FIGURE 2.7. Locating the ports in the back of the Commodore 64 to attach the printer interface

TEST PROGRAM

The following BASIC program can be used to test the interface:

```
10 OPEN4,4:CMD4
20 FOR I=1 TO 11
30 READ SPEC
40 PRINT CHR$(27)CHR$(SPEC);
50 PRINT "TESTING 1,2,3 TESTING"
60 NEXT I
70 CLOSE 4
80 END
90 DATA 77,80,14,52,15,69,53,49,50,70,64
```

The program output is:

```
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
```

For now do not worry about how the program functions. Just verify that the output you obtain is identical to the output shown above.

IBM PC, PC XT, PC*jr*, and Compaq

The addition of a Printer Adapter card enables the IBM PC® and PC XT® computers to drive an Epson printer. Figure 2.8 contains a picture of a Printer Adapter card. This card is installed by inserting it into one of the available expansion slots.

The addition of a Parallel Printer Attachment enables the IBM PC*jr*® to drive an Epson printer. The Compaq® computer requires no additional interface material to drive an Epson printer.

CABLE

To accomplish the installation of the printer, a cable is required. The cable must be an IBM Printer Cable. One end of this cable has a male Centronics connector attached to it. The other end of the cable has a DB-25 connector attached to it. Both of these connectors are asymmetrical. This fact forces the connection to be made correctly. Figure 2.9 shows an IBM Printer Cable.

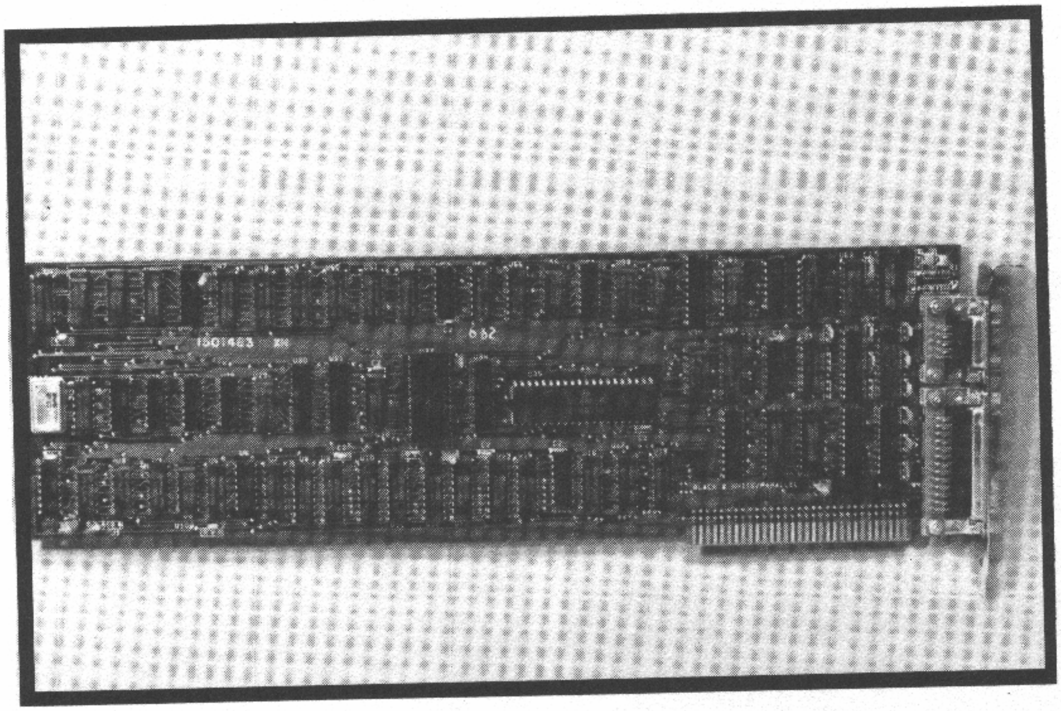
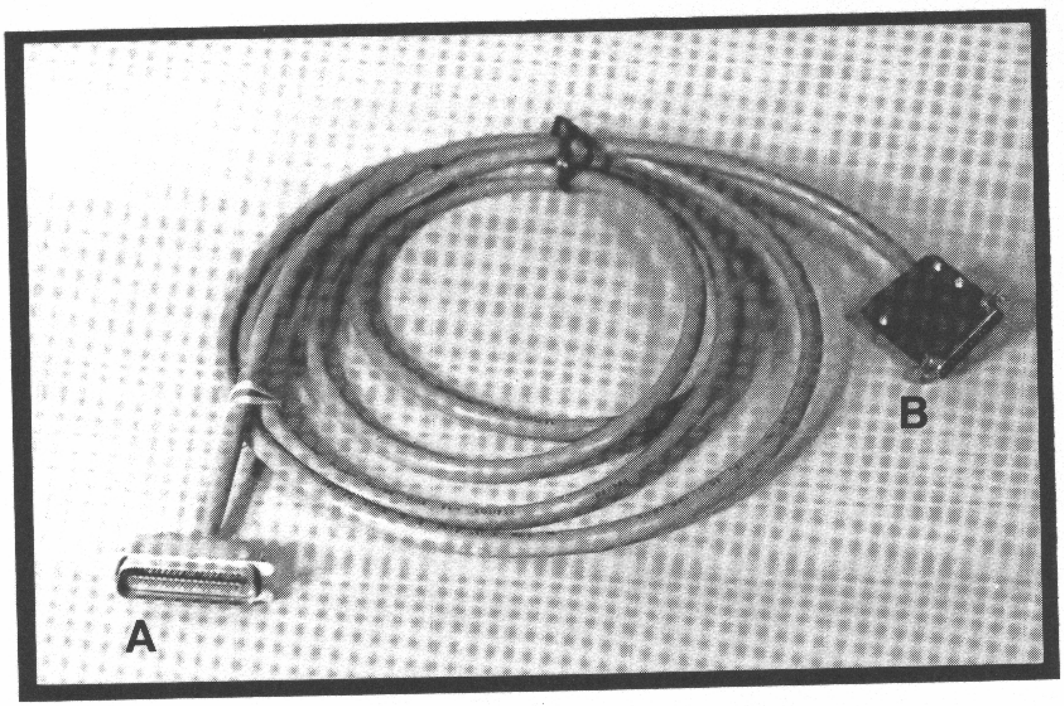


FIGURE 2.8. Additional interface material necessary for an IBM computer to drive an Epson printer



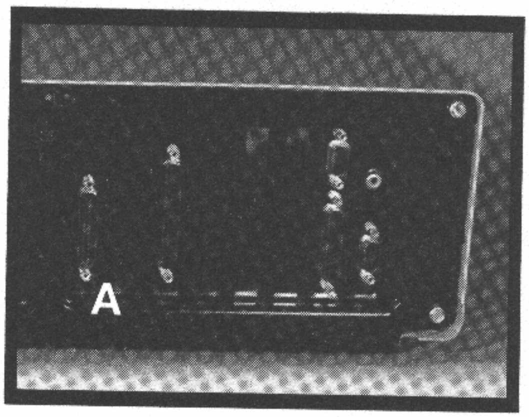
a. Centronics connector; *b.* DB-25 connector

FIGURE 2.9. An IBM Printer Cable

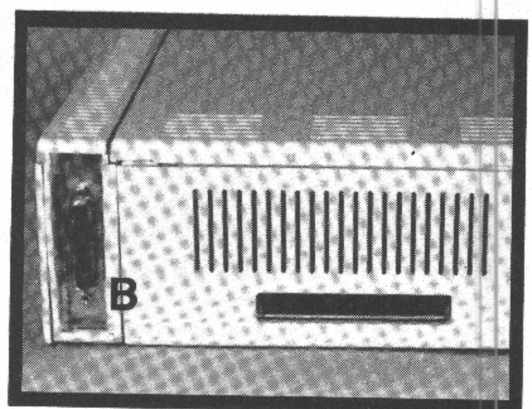
HOOK UP

Verify that both the printer and the computer are shut off before hooking the printer up to the parallel printer port.

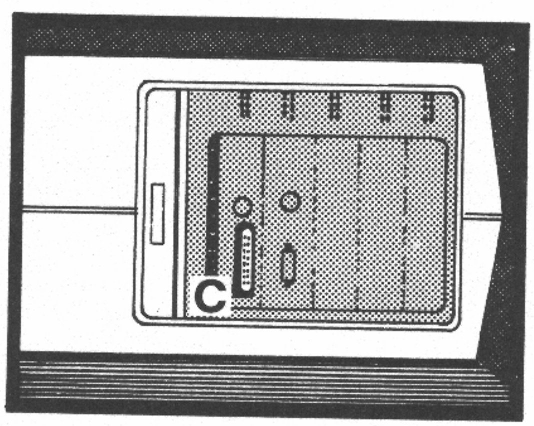
First, connect the end of the cable attached to the DB-25 connector to the parallel printer port located in the rear of the computer. Figure 2.10 shows the location of this port. Note that only one alignment can be used to make the connection. The wider side of the connector must be facing left.



a, IBM PC or PC XT parallel printer port



b, IBM PCjr parallel printer port



c, Compaq parallel printer port

FIGURE 2.10. Locating the printer port in an IBM or Compaq computer

Now connect the other end of the cable to the port in the back of the printer, as discussed at the beginning of this chapter.

TEST PROGRAM

The following BASIC program can be used to test the computer/printer interface:

```
10 FOR I=1 TO 11
20 READ SPEC
30 LPRINT CHR$(27)CHR$(SPEC);
40 LPRINT "TESTING 1,2,3 TESTING"
50 NEXT I
60 END
70 DATA 77,80,14,52,15,69,53,49,50,70,64
```

The program output is:

```
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
```

For now do not worry about how the program functions. Just verify that the output you obtain is identical to the output shown above.

Kaypro II and Sanyo MBC

The Kaypro II® and Sanyo MBC® personal computers have a parallel printer port as a standard feature. No additional interface material is required.

CABLE

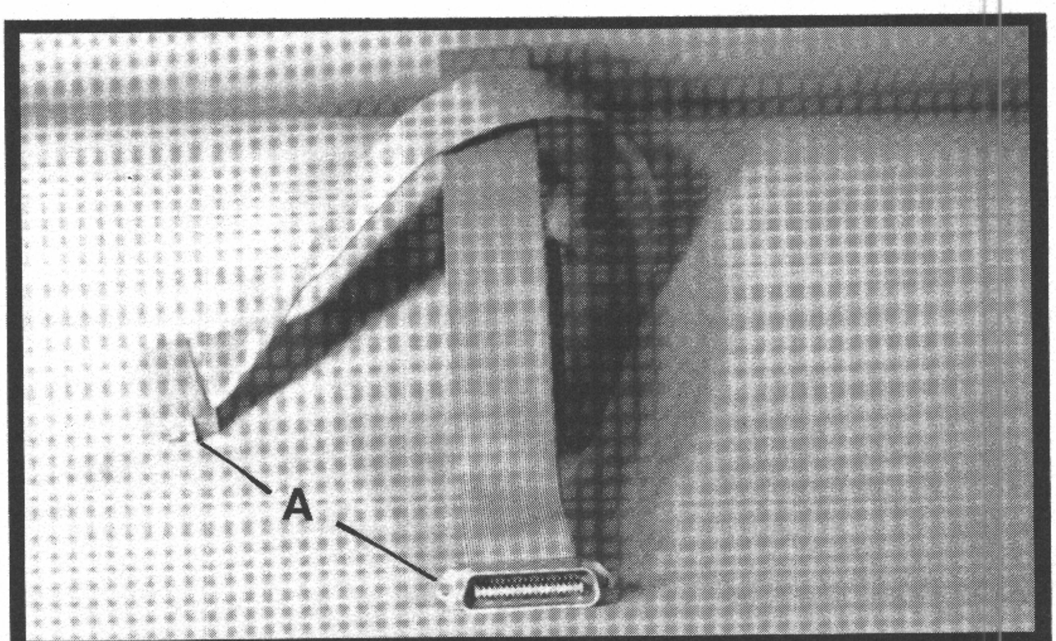
To accomplish the installation of the printer, a cable is required. The cable must be a parallel type cable. Each end of the cable must have a male Centronics connector attached to it. Centronics connectors are asymmetrical. This fact forces the connection to be made correctly. Figure 2.11 shows a typical ribbon cable with Centronics connectors.

HOOK UP

Verify that both the printer and the computer are shut off before hooking the printer up to the computer.

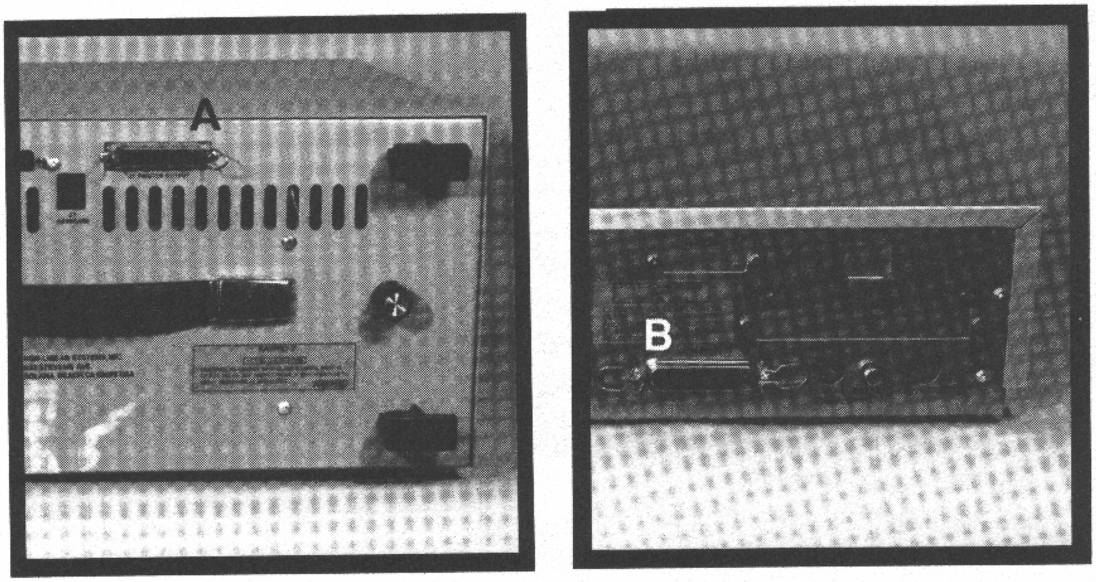
First connect one end of the cable to the parallel printer port located in the rear of the computer. The port is labelled PRINTER. Note that only one alignment can be used to make the connection. The wider side of the connector must be facing up. Figure 2.12 shows the location of the port in the computer.

Now connect the other end of the cable to the port in the back of the printer, as discussed at the beginning of this chapter.



a, Centronics connectors

FIGURE 2.11. Typical ribbon cable with Centronics connectors



a, Kaypro parallel printer port

b, Sanyo parallel printer port

FIGURE 2.12. Locating the parallel printer port in the back of the Kaypro or Sanyo computers

TEST PROGRAM

The following BASIC program can be used to test the computer/printer interface:

```
10 FOR I=1 TO 11
20 READ SPEC
30 LPRINT CHR$(27)CHR$(SPEC);
40 LPRINT "TESTING 1,2,3 TESTING"
50 NEXT I
60 END
70 DATA 77,80,14,52,15,69,53,49,50,70,64
```

The program output is:

```
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
TESTING 1,2,3 TESTING
```

For now do not worry about how the program functions. Just verify that the output you obtain is identical to the output shown above.

3

Epson Printers with Widely-Used Applications Software Packages

Introduction

In this chapter, we will discuss the steps involved in installing Epson printers on various software packages. We will provide the correct responses to the installation prompts for each package. We will also offer hints on how to utilize as many of the printer's features as possible. The subject of control codes is mentioned in several of the discussions. Chapters 4 through 11 include a discussion of the control codes for the RX and FX Series. Appendix B contains a complete listing of the Epson control codes.

dBASE II by Ashton Tate

In this section, we will discuss the use of Epson printers with the dBASE II® data base application package. For the purposes of this discussion, we will assume the computer hardware to be an IBM PC.

There are no references to the printer in the dBASE II installation program. The package can use any text printer. However, the dBASE II program does allow printer control codes to be sent. Any of the Epson's features can be enabled in this fashion.

Before sending the control codes, the printer must be enabled. To accomplish this task, type the following at the dBASE period prompt:

```
SET PRINT ON
```

The printer control codes are sent in a fashion similar to the BASIC expression for the control codes. But, CHR is used in place of CHR$. Also, all of the characters must be specified in ASCII code form. Each character must be separated by a plus sign (+). These constraints result in the code to enable the elite mode from dBASE II appearing as:

```
CHR(27) + CHR(77)
```

Compare this to the BASIC expression to enable the elite mode.

```
CHR$(27)"M"
```

Note that an "M" corresponds to an ASCII code of 77. Finally, a question mark must precede any occurrence of a printer control code. The command to enable the elite mode from dBASE II has the form:

```
? CHR(27) + CHR(77)
```

Other control codes can be sent to the printer in the same fashion. Use the ASCII table in Appendix A to accomplish the conversion of a character to an ASCII code.

A single command can be used to send more than one control code. The next code should be added to the end of the preceding code. Remember that a plus sign (+) must separate each character, specified by CHR. For example, the command to enable both the elite mode and the double-strike mode is:

? CHR(27) + CHR(77) + CHR(27) + CHR(69)

Lotus 1-2-3 by Lotus Development Corporation

In this section, we will discuss the steps involved in configuring the 1-2-3 and PrintGraph programs from the Lotus™ 1-2-3™ electronic spreadsheet package for use with an Epson printer.

Both of the Epson Series printers will function adequately without reconfiguring these programs. However, some of the parameters can be changed in order to suit personal tastes or needs. Note that for the purposes of this discussion, we assume the hardware to be an IBM XT.

RECONFIGURING THE 1-2-3 PROGRAM

The default configuration of 1-2-3 will function adequately with either the Epson RX or FX Series. However, you may want to make some minor changes such as changing the margin settings. The following discussion details how to accomplish such changes.

After entering the 1-2-3 program from the access system, we caused the menu to be displayed by striking the slash (/) key. We then typed "WGD." This action caused the deafult configuration menu to be shown on the screen. We selected the printer option by striking the Return key.

We elected to change the default left margin setting. To accomplish this task, we first moved the cursor until it rested upon "Left." The cursor can be moved to the right and left by the right and left arrow keys on the numeric keypad, respectively. Once the cursor was resting upon "Left," we struck the Return key. The screen displayed:

Default left margin (0..240): 4

We entered 0 and struck the Return key. The default left margin setting was changed to 0. We changed the right margin setting to 80 in a similar fashion.

The "Setup" option can be used to change the default setup string. This string is used to initialize the printer. We elected to use the compressed pitch mode. So, we entered the code for the compressed mode:

/015

and pressed the Return key. Now, the compressed pitch mode will be enabled whenever the printer is used by the package.

We then used the "Quit" option to return to the default configuration menu. We used the "Update" option to save these changes. We then used the "Quit" option to return to the 1-2-3 program. The reconfiguration of the 1-2-3 program is now complete.

Other printer features that can be changed by reconfiguring the 1-2-3 program are the type of interface, use of automatic line feed, the top and bottom margins, and the page length. All of these features are changed in a manner similar to the above examples.

RECONFIGURING THE PRINTGRAPH PROGRAM

The default configuration of PrintGraph will function adequately with either the Epson RX or FX Series. However, you may want to make a minor alteration such as changing the graphics density.

After entering the PrintGraph program from the access system, we chose the "configure" option. This option causes the PrintGraph default menu to be displayed. We selected the "Device" option from this menu. This option causes a menu of graphic output devices to be displayed. We selected the Epson FX80 printer with double density graphics. The FX selection can also be indicated for the RX Series. To accomplish this selection, we moved the cursor until it rested upon the desired device. The cursor can be moved up and down by using the up and down arrows on the numeric keypad. Once the cursor was in the correct position, the mark had to be moved. This action can be accomplished by striking the space bar. Now, striking the Return key will cause the marked device to be selected. This action will cause the PrintGraph default menu to be displayed once again.

We used the save option to make the change permanent. After selecting the save option, the screen will display the choices of "Cancel" and "Replace." Choose the "Replace" option. Following this selection, the screen will again display the PrintGraph default menu. The "Quit" option will cause the PrintGraph general menu to be displayed. The reconfiguration of the PrintGraph program is now complete.

pfs:file and pfs:report by Software Publishing Corporation

In this section, we will discuss the Printer program in the pfs:file® and pfs:report® packages. The packages will function adequately with no changes to the Printer program. But, if a different printer feature is desired, the printer setup characters can be defined. Note that for the purposes of this discussion, we will assume the computer hardware to be an IBM PC.

The printer program must be accessed at the system level. The procedure is the same for both packages. The following procedure must be observed to define a setup string:

```
A>PRINTER
```

After this program is loaded, the screen will display:

```
                    PRINTER SETUP

This program allows you to send special characters to a printer.

   Turn your printer off and on, and enter its name below.

                 PRINTER NAME: LPT1:

         (C) 1982 Software Publishing Corporation
                                              F10-Continue
```

LPT:1 is the correct name for a parallel printer attached to the parallel printer port. Turn the printer off and on as instructed. Strike the F10 key without changing the printer name. The screen will now display:

```
      Enter printer setup characters.  Press F10 when done.
            The characters will be sent to the printer.
   If you make a mistake, press F6 to escape from the program,
                   and then run the program again.
```

Any of the Epson's features may be enabled by the setup characters. To enable the elite pitch and italicized characters, enter the following:

```
        Enter printer setup characters.  Press F10 when done.
              The characters will be sent to the printer.
     If you make a mistake, press F6 to escape from the program,
                    and then run the program again.

: Escape
: M
: Escape
: 4
:
```

Note that "Escape" was input by pressing the control and escape keys simultaneously. Striking the F10 key will cause the setup characters to be stored and an exit from the program to the system level to be executed. All pfs:file output will now be generated using elite sized, italicized characters.

Symphony by Lotus Development Corporation

In this section we will discuss how to install an Epson FX or RX printer so that it can be used with the Symphony® package. The installation program must be run before the package can be used. Note that for the purpose of this discussion, we will assume the computer hardware to be an IBM PC.

The installation program must be accessed from the operating system. Insert the copy of Install Program Disk into drive A. Follow the procedure below to accomplish the installation.

```
A>INSTALL
```

This will cause the installation program to be loaded. The screen will display:

```
                    Symphony Install

           Copyright (c)  1982, 1983, 1984
           Lotus Development Corporation
                 All Rights Reserved

                    Version 0.11

TWO-DISKETTE SYSTEM: Remove the Install Program Disk and insert
the Install Library Disk. Then press any key to continue or press
[BREAK] to quit.

HARD DISK SYSTEM: Before proceeding, copy the contents of the
Install Library Disk into the directory where you copied the
contents of the Install Program Disk. Press [BREAK] to quit.
```

After we loaded the Install Library Disk into drive A and pressed any key, the screen displayed:

```
      For Symphony and its companion programs to work with your
      computer system, you must provide information about the
      equipment that you are using. The Install program assembles
      this information about your hardware setup in a special file
      called a driver set.

      Please be sure that you have completed the Hardware Chart
      at the beginning of Chapter 3 in the "Introduction to
      Symphony." You need this information during installation.

      Before you begin, you can press [HELP] if you would like more
      information about installation or important keys you use with
      the Install program.
--------------------------------------------------------------------------

      Press [HELP] for more information.
or    Press [SPACE] to begin.
```

We pressed the space key to begin the installation. The screen then displayed the main menu:

```
--------------------------------------------------------------------------
                          M A I N   M E N U
--------------------------------------------------------------------------

     1   Installation: Create a driver set

     2   Display selections in a driver set

     3   List names of driver sets

     4   Change selections in a driver set

     5   Exit Install program

--------------------------------------------------------------------------
         Type 1 - 5 and press [RETURN].
   or    Press [HELP] for explanation of menu options.  1
```

Note that our selection is circled. We chose option 1 so that we could define a set of drivers. The following message was then displayed on the screen:

```
         To create a driver set, you must complete several steps.
         The next screens explain these steps:

         *  Add single drivers to the Lotus master library. This
            step is OPTIONAL.

         *  Assign a name to the driver set or use the default
            name.

         *  Make selections from menus that are appropriate to
            your computer system.

         *  Save the driver set on your hard disk or on the
            Symphony disks if you have a two-diskette system.

--------------------------------------------------------------------------
         Press [SPACE] to go to first step.
   or    Press [HELP] for more information.
```

We pressed the space bar to begin creating a driver set. We now skip to the portion of the installation program pertaining to selection of printer drivers. At that point, the screen will display:

```
      If you have a printer and want to print copies of the work
      you do with Symphony, you must select a text printer driver.

      Do you want to select a text printer ?

--------------------------------------------------------------------

      Type Y and press [RETURN] to select text printer.
or    Press [SPACE] to select next driver type.  Y
```

We wanted to specify a text printer. The answer of "Y" caused the screen to display the following menu of text printers:

```
Selecting a Text Printer driver is OPTIONAL.
--------------------------------------------------------------------

1        Anadex 9620A Silent Scribe
2        C. Itoh 8510A
3        Canon A1210 or Quadram Quadjet
4        DEC LA100
5        Diablo 630
6        Diablo Series C
7        Epson FX, RX and JX series
8        Epson MX series
9        GE/Genicom 3000, B&W/color
10       Gemini Star 10X or 15X
11       Generic - forced auto-LF
12       Generic - full capability
13       Generic - no backspace
14       Generic - no backspace, forced auto-LF
15       HP 2225, 2930, or 2686
--------------------------------------------------------------------
      Type 1 - 31 and press [RETURN] to enter selection.
or    Press [SPACE] to view more selections.
or    Press [HELP] for more information.  7
```

We selected option 7 to install an Epson RX or FX into the package as a text printer. The screen then displayed the following prompt:

```
If you've selected a Text Printer driver, it's marked *.
--------------------------------------------------------------------------
1         Anadex 9620A Silent Scribe
2         C. Itoh 8510A
3         Canon A1210 or Quadram Quadjet
4         DEC LA100
5         Diablo 630
6         Diablo Series C
7    *    Epson FX, RX and JX series
8         Epson MX series
9         GE/Genicom 3000, B&W/color
10        Gemini Star 10X or 15X
11        Generic - forced auto-LF
12        Generic - full capability
13        Generic - no backspace
14        Generic - no backspace, forced auto-LF
15        HP 2225, 2930, or 2686
--------------------------------------------------------------------------
      Press [RETURN] to confirm marked selection.
or    Type 1 - 31 and press [RETURN] to enter new selection.
or    Press [SPACE] to view more selections.
```

We pressed the Return key to complete the selection of an RX or FX printer driver. The screen then displayed:

```
      If you have a graphics printer or plotter and want to print
      the graphs that you create with Symphony, you must select a
      graphics printer or plotter driver. Even if you selected your
      printer from the text printer menu, if it can print graphs
      and you want to use the printer with Symphony, you must also
      select it from the graphics printer and plotter menu.
      You can select as many of these drivers as you wish, but
      Lotus suggests a maximum of four because of file size
      limitations for driver sets.
      Do you wish to select a graphics printer or plotter ?
--------------------------------------------------------------------------
      Type Y and press [RETURN] to select graphics printer.
or    Press [SPACE] to select next driver type.  Y
```

Since we wanted to select a graphics printer, we answered with "Y". This response caused the following menu of graphics printers to be displayed:

```
Selecting a Graph Printer driver is OPTIONAL.
------------------------------------------------------------------------
1        Amdek Amplot II
2        Anadex 9620A Silent Scribe
3        Canon A1210 or Quadram Quadjet
4        DEC LA100
5        Diablo Series C
6        Enter Sweet P Plotter
7        Epson FX and RX series, density 1
8        Epson FX and RX series, density 2
9        Epson FX and RX series, density 3
10       Epson FX and RX series, density 4
11       Epson JX-80, density 1
12       Epson JX-80, density 2
13       Epson JX-80, density 3
14       Epson JX-80, density 4
15       Epson LQ-1500, density 1

------------------------------------------------------------------------
      Type 1 - 53 and press [RETURN] to enter selection.
or    Press [SPACE] to view more selections.
or    Press [HELP] for more information.  8
```

We chose option 8. This option selects the printer driver for an RX or FX printer using double density graphics. The screen then displayed the following prompt:

```
If you've selected a Graph Printer driver, it's marked *.
----------------------------------------------------------------------
1          Amdek Amplot II
2          Anadex 9620A Silent Scribe
3          Canon A1210 or Quadram Quadjet
4          DEC LA100
5          Diablo Series C
6          Enter Sweet P Plotter
7          Epson FX and RX series, density 1
8    *     Epson FX and RX series, density 2
9          Epson FX and RX series, density 3
10         Epson FX and RX series, density 4
11         Epson JX-80, density 1
12         Epson JX-80, density 2
13         Epson JX-80, density 3
14         Epson JX-80, density 4
15         Epson LQ-1500, density 1
----------------------------------------------------------------------

      Type 1 - 53 and press [RETURN] to enter additional selection.
or    Press [SPACE] to view more selections.
or    Press [RETURN] to select next driver type.
```

We pressed the Return key to complete the selection.

The specification of the printer drivers is now finished. Be sure to run the installation program to completion and save the driver set on the disks.

WordStar by MicroPro

In this section, we will discuss the use of Epson printers with the WordStar® word processing package. For the purposes of this discussion, we will assume the computer hardware to be an IBM PC. First, we will discuss the portion of the WordStar installation program pertaining to printer specifications.

STARTING THE INSTALLATION

To start the installation program, type "WINSTALL" at the system prompt. After the copyright messages, the main installation menu will appear:

```
        *****  INSTALLATION MENU  *****

If you are installing a new copy of WordStar, you must select
letter A to install your terminal, then letter C to install
your printer. If your terminal is not listed on the Menu of
Terminals, return to this menu and select letter B. If your
printer is not listed on the Menu of Printers, return to this
menu and select letter D. If you want to change a particular
WordStar feature, choose letter E.


A  Menu of Terminals
B  Custom Installation of Terminals
C  Menu of Printers
D  Custom Installation of Printers

E  Menu of WordStar Features

X  Exit from INSTALL

   Enter the letter of your choice (A/B/C/D/E/X).D
```

Note that we chose option D. The FX and RX printers do not appear on the Menu of Printers. We will carry out a custom installation. The following message will appear on the screen:

```
          *****   PRINTER INSTALLATION MENU   *****

     If your printer was not listed on the menu of printers or,
     if you want to enhance the performance of your printer, you
     must provide some additional information about your printer.
     This information should be in the manual that comes with the
     printer.  If not talk to your dealer.

     If you wish to install a specialty printer select A on the
     following menu.

     If you wish to install a standard printer, select B on the
     following menu.

     If you just want to change a feature, select the appropriate
     letter on the following menu.




     Type any key to continue...
```

Striking any key will cause the actual printer installation menu to appear on the screen:

```
      *****   PRINTER INSTALLATION MENU   *****

A  Automatic installation for Specialty printers
B  Automatic installation for Standard printers

All printers                          Specialty printers only

C  Printer name                       I  Ribbon selection
D  Initialization                     J  Vertical motion
E  Overprinting                       K  Horizontal motion
F  Boldfacing                         L  Print modes
G  Protocol menu                      M  Phantom characters
H  Driver menu

Standard printers only                Optional

N  Return / line feed                 O  User-defined functions
                                      P  Carriage roll
                                      Q  Character pitch
X  Exit to INSTALLATION menu

   Enter the letter of your choice (A-Q/X).
```

We selected option A because we want to change some of the items on the specialty printers list. This choice causes the following message to appear:

```
       Automatic installation of a specialty printer

   This routine allows you to install a specialty printer
   without returning to the printer installation menu.

   Once you enter this routine you will have to answer questions
   on each printer feature.

       Enter "X" to exit,
       or press <RETURN> to continue. <RETURN>
```

AUTOMATIC PROMPTS

The following prompts will now appear, one after another. The parameters specified by these prompts allow us to make more complete use of Epson printers' many features. We answered the prompts in the following fashion:

```
         Printer Name (optional)

  Current name is :        IBM Parallel Printer


     Enter "C" to change,
     or press <RETURN> to leave unchanged. C



     New printer (maximum 34 characters) : EPSON RX/FX


 Printer name now is : EPSON RX/FX


    If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

"Epson RX/FX" will now appear in place of "IBM Parallel Printer"* in the copyright message when WordStar is run. <RETURN> indicates that we pressed the Return key.

* The current printer name may be different for your version of WordStar.

Another series of individual prompts will be displayed. We answered these prompts as follows:

```
                    Printer initialization

These are the sequences transmitted to the printer at the
beginning and conclusion of printing.


Printer initialization function code sequence is currently : Dh


Printer de-initialization function code sequence is currently :  (empty)



    Enter "C" to change,
    or press <RETURN> to leave unchanged. <RETURN>
```

Note that a hexadecimal D represents a carriage return for Epson printers. This insures that the first line of output will begin at the left margin of a line.

We answered the next prompt as shown:

```
                         Overprinting

         What type of backspacing method does your printer use?

         Currently : Non-backspacing standard printer



             Enter "C" to change,
             or press <RETURN> to leave unchanged. C
```

We want to change this parameter because Epson printers are capable of backspacing. This response caused the following prompts to appear which we answered as indicated.

```
                    Overprinting

    What type of backspacing method does your printer use?

    Currently : Non-backspacing standard printer



    A   Specialty printer
        Printer is a daisy wheel or similar incremental printer
          and can overprint using horizontal motion controls

    B   Backspacing standard printer
        Printer can overprint by backspacing as well as by
          using return without a line feed

    C   Non-backspacing standard printer
        Printer can overprint only by using return without
          a line feed

        Enter the letter of your choice (A/B/C),
        or press <RETURN> to leave unchanged. B

  Now : Backspacing standard printer

    If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

We chose the B option because Epson printers are backspacing standard printers.

```
You must now enter the backspacing control sequence.

Backspacing function code sequence is currently : 8h


    Enter "C" to change,
    or press <RETURN> to leave unchanged. <RETURN>


Backspacing function code sequence is now : 8h


    If this is correct, enter Y or <RETURN>. If not, enter N. <RETURN>
```

A hexadecimal 8 represents a backspace to Epson printers. So, we left the backspacing function code unchanged. The next set of prompts were answered in the following fashion:

```
                         Boldfacing

        You may set the number of strikes for boldfacing. This
        should be set to 2 for specialty printers.



        Boldfacing strikes value is currently 2.



            Enter "C" to change,
            or press <RETURN> to leave unchanged. <RETURN>
```

Note that a different number of strikes could have been selected. We think that two overstrikes is quite adequate.

The response to the next prompt is shown below:

```
Communications protocol is currently :  No protocol


     Enter "C" to change,
     or press <RETURN> to leave unchanged. <RETURN>
```

The IBM hardware will handle the protocol. It is not necessary that the protocol be embedded in the software.

The next prompt was answered as shown below:

```
              Printer driver


   As there are a large number of values relating to this item,
   these values are not displayed. If you wish to change or
   view any of the current values enter C below and each item
   will be displayed individually.


       Enter "C" to change,
       or press <RETURN> to leave unchanged. <RETURN>
```

Note that this response specifies a parallel printer driver. The parallel printer driver is the default driver.

The next set of prompts deals with ribbon selection. Since Epson RX and FX Series printers have only one ribbon, we used the ribbon selection

to specify the selection of the alternate or regular character set. Note that any other feature that can be enabled and deactivated could be used in this sequence. Examples include the underlined mode and unidirectional printing. Our responses are shown below:

```
                 Ribbon selection


With these control sequences you specify alternate and
standard ribbon selection, if your printer allows. What
sequence of characters should be sent to the printer at
"^PY" in your text?


Alternate ribbon selection function code sequence is currently :  (empty)


Standard ribbon selection function code sequence is currently :  (empty)


    Enter "C" to change,
    or press <RETURN> to leave unchanged.C
```

```
            Alternate ribbon selection

With this control sequence you specify alternate ribbon
selection, if your printer allows. What sequence of characters
should be sent to the printer at the first "^PY" in your text?

Alternate ribbon selection function code sequence is currently :  (empty)


    Enter "C" to change,
    or press <RETURN> to leave unchanged.C
```

Figure 3.1 contains a message pertaining to the input of a function code sequence. This message will appear on the screen.

```
You can enter a value in ASCII, decimal, or hexadecimal
codes. Precede each entry with these prefixes:

ASCII          :    (:^A enters ASCII ^A, a single value)
Hexadecimal    ,    (,41 enters hexadecimal 41)
Decimal        #    (#61 enters decimal 61)

To enter a sequence of characters, enter each one separately,
followed by <RETURN>.

Press <RETURN> to leave a value unchanged.
Enter a period (.) and press <RETURN> to terminate a sequence
    and to eliminate all subsequent values.

These special characters require hexadecimal input:
<RETURN> (^M)              ,0D
Period (.)                 ,2E
^H (backspace: ^H)         ,08
```

FIGURE 3.1. Message explaining how to input a function sequence

The following prompt will appear after the message. The program expects a control code to be input.

```
Maximum entries for alternate ribbon selection function is 4.

Current      New
Value        Value
00h          ,1B
00h          :4
00h          ,0
00h          ,0


Alternate ribbon selection function code sequence is now : 1Bh 34h 0h 0h


   If this is correct, enter Y or <RETURN>. If not, enter N. <RETURN>
```

We answered the prompt as shown. A hexadecimal 1B represents the escape character. The decimal 4 enables the italic character set. We responded to the next prompt as indicated:

```
                Standard ribbon selection

With this control sequence you specify standard ribbon
selection. What sequence of characters should be sent to
the printer at the second "^PY" in your text?



Standard ribbon selection function code sequence is currently :  (empty)




   Enter "C" to change,
   or press <RETURN> to leave unchanged. C
```

The message, contained in figure 3.1, pertaining to the input of a function code sequence, will now appear. We answered the prompt following this message as shown below:

```
Maximum entries for standard ribbon selection function is 4.

Current     New
Value       Value
00h          ,1B
00h          :5
00h          ,0
00h          ,0


Standard ribbon selection function code sequence is now : 1Bh 35h 0h 0h


    If this is correct, enter Y or <RETURN>. If not, enter N. <RETURN>
```

Note that the alternate ribbon selection in WordStar will now actually select the alternate character set. The standard ribbon selection in WordStar will enable the standard character set.

We responded to the next prompt as shown:

```
                     Vertical motion


      As there are a large number of values relating to this item,
      these values are not displayed. If you wish to change or
      view any of the current values enter C below and each item
      will be displayed individually.


          Enter "C" to change,
          or press <RETURN> to leave unchanged. <RETURN>
```

The vertical motion parameters are specified in terms used by daisy wheel printers. The Epson printers cannot use this scheme, so we skipped this section. We responded to the following prompt as indicated:

```
              Horizontal motion

As there are a large number of values relating to this item,
these values are not displayed. If you wish to change or
view any of the current values enter C below and each item
will be displayed individually.


   Enter "C" to change,
   or press <RETURN> to leave unchanged. <RETURN>
```

Epson printers cannot move in an incremental fashion on the horizontal. So, we skipped these parameters.

We answered the next prompt as shown:

```
              Print modes


If your printer is capable of both backward and forward
printing set both of the following features.

Set forward print mode function code sequence is currently : 1Bh 35h

Set backward print mode function code sequence is currently : 1Bh 36h


   Enter "C" to change,
   or press <RETURN> to leave unchanged. <RETURN>
```

Epson printers do not use this scheme of printing. It is more typical of daisy wheel printers. So, we skipped these parameters. The final automatic prompt we answered as indicated below:

```
            Print phantom characters

These sequences specify the characters to be sent to the
printer to print its phantom space and phantom rubout
characters.


Print phantom space function code sequence is currently : 1Bh 59h


Print phantom rubout function code sequence is currently : 1Bh 5Ah



    Enter "C" to change,
    or press <RETURN> to leave unchanged. <RETURN>
```

Epson printers do not have phantom characters. So, we ignored this set of parameters.

USER-DEFINED FUNCTIONS

The printer installation menu will now reappear. Notice that option O allows the user to define functions. For the purpose of this example, we will specify O in order to define several of the Epson's features as user-defined functions. Any of the features available on Epson printers can be defined as user functions. We chose what we thought to be four of the more useful features.

When O is specified, the following messages will appear:

```
                User-defined functions


WordStar provides four print commands which the user may
assign. When you include function characters (such as ^PW)
in your text, WordStar will perform actions, such as
changing to italic font. Check your printer manual for
special features available.


                User function #1

What sequence of characters should be sent to the printer
at "^PQ" in your text?

#1 User function code sequence is currently :  (empty)



    Enter "C" to change,
    or press <RETURN> to leave unchanged. C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. We answered the prompts following the message as shown:

```
Maximum entries for #1 user function is 4.

Current      New
Value        Value
00h          ,E
00h          ,0
00h          ,0
00h          ,0

#1 User function code sequence is now : Eh 0h 0h 0h

If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

This definition allows the expanded print mode to be enabled for one line only. In other words, when the code for user-defined function #1, ^ PQ, is included in the text, the Epson will enable the expanded print mode for one line.

The prompt for defining user function #2 now appears:

```
                User function #2

What sequence of characters should be sent to the printer
at "^PW" in your text?

#2 User function code sequence is currently :  (empty)



   Enter "C" to change,
   or press <RETURN> to leave unchanged. C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. We answered the prompts following the message as shown:

```
Maximum entries for #2 user function is 4.

Current     New
Value       Value
00h          ,1B
00h          :T
00h          ,0
00h          ,0

#2 User function code sequence is now : 1Bh 54h 0h 0h

   If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

This definition for user function #2 allows the script modes to be deactivated when ^ PW is entered in the text. The prompt for defining user function #3 now appears:

```
                User function #3

What sequence of characters should be sent to the printer
at "^PE" in your text?

#3 User function code sequence is currently :  (empty)

    Enter "C" to change,
    or press <RETURN> to leave unchanged. C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. We answered the prompts following the message as shown:

```
Maximum entries for #3 user function is 4.

Current      New
Value        Value
00h          ,F
00h          ,0
00h          ,0
00h          ,0


#3 User function code sequence is now : Fh 0h 0h 0h


    If this is correct, enter Y or <RETURN>. If not, enter N. <RETURN>
```

This definition for user function #3 allows the compressed print mode to be enabled when ^ PE is included in the text. The prompt for defining user function #4 now appears:

```
                User function #4

        What sequence of characters should be sent to the printer
        at "^PR" in your text?

        #4 User function code sequence is currently :  (empty)



           Enter "C" to change,
           or press <RETURN> to leave unchanged.C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. We answered the prompts following the message as shown:

```
   Maximum entries for #4 user function is 4.

   Current     New
   Value       Value
   00h          ,12
   00h          ,0
   00h          ,0
   00h          ,0

   #4 User function code sequence is now : 12h 0h 0h 0h


      If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

This definition for user function #4 allows the compressed print mode to be deactivated when ^ PR is included in the text.

DEFINING THE SCRIPT MODES

The printer installation menu will appear once again. This time we chose option P. Option P is used to define the amount of paper movement used to output superscripts and subscripts. Instead, we will define the two script modes to accomplish this task. Note that you will have to use the second user-defined function (^PW) to deactivate either of the script modes. Also, you will have to manually deactivate the script mode after it has been used automatically by WordStar, such as in the automatic page numbering feature.

Finally, the example file, PRINT.TST, will no longer print correctly if you make these changes. We think this is a small price to pay for the increased visual appeal of true superscripts and subscripts.

We answered this set of installation menu prompts as follows:

```
                Roll up carriage

     Some WordStar features, such as superscripting, require that
     the carriage roll up a partial line. Check your printer
     manual for the special characters required to roll the
     carriage up a partial line. What sequence of characters
     should be sent to the printer at "^PT" in your text?

     Roll up carriage function code sequence is currently :  (empty)


        Enter "C" to change,
        or press <RETURN> to leave unchanged.C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. The following prompt will appear after the message:

```
Maximum entries for roll up carriage function is 4.

Current     New
Value       Value
00h          ,1B
00h          :S
00h          :0
00h          ,0


Roll up carriage function code sequence is now : 1Bh 53h 30h 0h

    If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

The occurrence of ^ PT in the text will now activate the superscript mode.

The next prompt to appear on the screen will be:

```
              Roll down carriage

   Some WordStar features, such as subscripting, require that
   the carriage roll down a partial line. Check your printer
   manual for the special characters required to roll the
   carriage down a partial line. What sequence of characters
   should be sent to the printer at "^PV" in your text?

   Roll down carriage function code sequence is currently :  (empty)



       Enter "C" to change,
       or press <RETURN> to leave unchanged. C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. The following prompt will appear after this message:

```
Maximum entries for roll down carriage function is 4.

Current     New
Value       Value
00h          ,1B
00h          :5
00h          :1
00h          ,0


Roll down carriage function code sequence is now : 1Bh 53h 31h 0h


    If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

The occurrence of ^ PV in the text will now activate the subscript mode.

ALTERNATE PITCH

The printer installation menu will appear again. We will select Q as our next option. Option Q is used to define an alternate character pitch. We selected the Epson's elite pitch as the alternate character pitch.

```
        Set alternate character pitch

If your printer allows, you can change the character pitch
within a document. This sequence specifies the characters
required to set alternate character pitch. Check your printer
manual for code sequence. What sequence of characters should
be sent to the printer at "^PA" in your text?

Set alternate character pitch function code sequence is currently :  (empty)




   Enter "C" to change,
   or press <RETURN> to leave unchanged. C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. The following prompt will appear after the message:

```
Maximum entries for set alternate character pitch function is 4.

Current     New
Value       Value
00h          ,1B
00h          :M
00h          ,0
00h          ,0


Set alternate character pitch function code sequence is now : 1Bh 4Dh 0h 0h


    If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

WordStar's alternate character pitch will now be Epson's elite pitch. In other words, when ^ PA is included in the text, subsequent printer output will be generated in elite pitch. The elite pitch includes 12 characters per inch.

The next prompt to appear is:

```
        Set standard character pitch

This code sequence provides the characters required to
set/reset standard character pitch. Check your printer manual
for the specific characters. What sequence of characters
should be sent to the printer at "^PN" in your text?

Set standard character pitch function code sequence is currently :  (empty)


   Enter "C" to change,
   or press <RETURN> to leave unchanged. C
```

The message pertaining to the input of a function code will now appear. Figure 3.1 contains a copy of this message. The following prompt will appear after the message:

```
Maximum entries for set standard character pitch function is 4.


Current     New
Value       Value
00h          ,1B
00h          :P
00h          ,0
00h          ,0

Set standard character pitch function code sequence is now : 1Bh 50h 0h 0h


    If this is correct, enter Y or <RETURN>. If not, enter N.  <RETURN>
```

Now the occurrence of ^ PN in the text will cause subsequent printer output to be generated using the pica pitch. The pica pitch includes 10 characters per inch.

EXITING THE INSTALLATION PROGRAM

The printer installation menu will appear one last time. Select the X option. This option causes the installation menu that appears on page 66 to be displayed on the screen. Choose the X option again. This choice will cause the following prompts to be displayed:

```
The changes made during this session of INSTALL are stored
in a temporary file. You may now save these changes in your
installed file WS.COM.

These are your current values:

Terminal                :       IBM Personal Computer
Printer                 : EPSON RX/FX
Communications protocol :           No protocol
Driver                  :       Parallel printer driver

          *****  EXIT OPTIONS MENU  *****

A   Save the changes made during this INSTALL session
B   Quit this session of INSTALL without saving changes
C   Change any of your choices / Remain in INSTALL

    Enter the letter of your choice (A/B/C).
```

Select the A option to save the changes made during the INSTALL session. The final messages will then appear indicating the end of the installation procedure:

```
Your new installed WordStar file is WS.COM.

You are returning to the operating system.

A>
```

INSTALLATION TEST

We tested the installation with the following text file:

```
^QTEST FILE
^ECONDENSED^R PICA^A ELITE^N PICA
^YITALICS^Y ROMAN
^VSUB ^V^TSUPER ^T^WNORMAL
```

When printed by WordStar, the file appears as follows. Note the effect of the commands used.

```
TEST FILE
CONDENSED PICA ELITE PICA
ITALICS ROMAN
SUB SUPER NORMAL
```

Note that the ^ PV and ^ PT commands must still be used in pairs. The WordStar package expects these two commands to cause some paper movement. The first occurrence of the command usually moves the paper to the position of the desired script. The second occurrence usually causes the paper to be repositioned to the site of the original print line. To avoid confusion, always use the ^ PV or ^ PT commands in pairs. Then, use the ^ PW command to return to the normal, nonscript mode.

4

Overview of Printer Modes

Introduction

This chapter deals with the fundamentals of using BASIC commands to control the printer. Chapters 5 through 11 will deal with the specific instructions used to control the printer.

In this chapter, the concepts of print modes and control codes will be defined. In the chapter, we will also explain how certain BASIC instructions work with the printer as well as how data is controlled within the printer. Finally, the procedure for resetting modes using a BASIC command will be detailed.

Printer Modes and Control Codes

A printer mode is a method of operation used by the printer. An example of a printer mode is the enlarged print mode. In the enlarged print mode, all data is output in type that is larger than normal.

```
This is the normal print mode.
Enlarged print mode
```

A control code is a message that is sent to the printer. The message will generally inform the printer which mode to select. A control code is never printed.

In the preceding example, an SO control code was sent to the printer. The SO control code carried the message, "Change to the enlarged print mode". This message was sent after the first line of data but before the second one. The SO control code caused the printer to change modes.

The control codes are detailed in appendix B. Notice from the table that these codes do not actually output characters. They merely change the print mode. Keep this important concept in mind when using control codes to change the print mode.

Setting a Mode from BASIC

VARIATIONS AMONG BASIC VERSIONS

In this book, we use the Microsoft version of BASIC for all program examples. Generally, the programs will work for other versions of BASIC with little or no alteration. For details of the version of BASIC installed on your computer, refer to your computer manual.

If your keyboard is not equipped to send a lowercase letter, look up the ASCII code value that corresponds to the desired lowercase letter and use the CHR$ function to send it. We discuss details of the ASCII code table and the CHR$ function later in this chapter.

Finally, all the control characters referred to in this book have two corresponding ASCII codes. We always use the lower of the two code values. If the code value specified by us does not work with your system, try using the higher code value.

ASCII CHARACTERS

A computer cannot actually store characters in its memory. Instead, a computer stores numeric values.

Before characters can be stored, they must be converted to numbers. Computers use special numeric codes to store characters. Most microcomputers use a code known as ASCII (American Standard Code for Information Interchange).

Epson printers also use the ASCII code. They have an ASCII code table in their memory. This code table is listed in appendix A.

All data received by the printer is encoded. The printer uses its own version of the ASCII code table to convert the numeric code to a character. This conversion is undertaken, one character at a time, while the data is being printed. For example, the numeric code value for the letter "A" is 65. Whenever the printer encounters a numeric code of 65 in the print data, an "A" will be output.

LLIST

The LLIST command is used in BASIC to produce a permanent listing of the program presently in memory. The form of the command is as follows:

LLIST

For example, suppose the following program was currently stored in memory:

```
5   LPRINT CHR$(27);"W";CHR$(1);
10 FOR I= 0 TO 7
20 LPRINT CHR$(27);"R";CHR$(I)
30 FOR J= 1 TO 12
40 READ A
50 LPRINT CHR$(A);"   ";
60 NEXT J
70 LPRINT
71 RESTORE
80 NEXT I
90 END
100 DATA 35,36,64,91,92,93,94,96
110 DATA 124,125,126
120 END
```

Issuing a LLIST command would result in the following output:

```
5   LPRINT CHR$(27);"W";CHR$(1);
10 FOR I= 0 TO 7
20 LPRINT CHR$(27);"R";CHR$(I)
30 FOR J= 1 TO 12
40 READ A
50 LPRINT CHR$(A);"   ";
60 NEXT J
70 LPRINT
71 RESTORE
80 NEXT I
90 END
100 DATA 35,36,64,91,92,93,94,96
110 DATA 124,125,126
120 END
```

Notice that the program is printed exactly as it appeared on the screen.

LPRINT, CHR$, and ESC

The LPRINT command is used in BASIC to send data to the printer. An LPRINT command with data specified causes that data to be output on the current line. After the data has been printed, a carriage return and a line feed will be output. These position the print head at the beginning of the next line.

```
10 LPRINT "THE FIRST LINE"
20 LPRINT "THE SECOND LINE"
30 END
```

The program output is:

```
THE FIRST LINE
THE SECOND LINE
```

Notice that execution of the LPRINT command in line 10 resulted in the first line of output. In addition, the LPRINT command in line 10 forced the print head to be repositioned at the beginning of the second line of output. The LPRINT command in line 20 caused the same action as the LPRINT command in line 10 with the exception that the output data was different. At the end of program execution, the print head was positioned at the beginning of what would have been the third line of output.

An LPRINT command with data specified and a trailing semicolon (;) causes output of the data only. The print head remains in the position immediately following the last character that was output.

```
10 LPRINT "THE FIRST LINE";
20 LPRINT " NOT THE SECOND LINE"
30 END
```

The program output is:

```
THE FIRST LINE NOT THE SECOND LINE
```

The LPRINT command in line 10 caused "THE FIRST LINE" to be output. The LPRINT command on line 20 caused "NOT THE SECOND LINE" to be printed. After line 20, the print head was positioned at the beginning of the line beneath the output. Compare this example with the preceding example.

An LPRINT command with no data specified will cause a carriage return and a line feed. When the LPRINT command is used in this fashion, it is often referred to as a blank LPRINT command.

```
10 FOR I=1 TO 10
20 LPRINT "A ";
30 NEXT I
40 LPRINT
50 LPRINT "THE LOOP PRINTED 10 A'S"
60 END
```

The program output is:

```
A A A A A A A A A A
THE LOOP PRINTED 10 A'S
```

There are several points to note in this program. First, the LPRINT command on line 20 resulted in ten letter A's being printed in the first line of output. If the semicolon had been omitted from this statement, ten lines each with a single letter "A" would have been printed in place of the one line. Also, the blank LPRINT command in line 40 caused the print head to be positioned at the beginning of the next output line. The LPRINT command in line 50 caused the second line of output. Without the blank LPRINT command, the second line of output would have been printed immediately following the ten A's on the first print line.

The CHR$ function in BASIC can be used with the LPRINT command to send an ASCII code to the printer. The configuration is:

LPRINT CHR$(*b*)

where *b* is a number or numerical expression from 0 to 255. Recall that the ASCII code for "A" is 65. Issuing the command:

LPRINT CHR$(65)

would result in an "A" being output. Note that the printer received the numeric code 65 and converted it so as to output an "A".

More significantly, the CHR$ function can be used to send control codes to the printer. Recall the SO control code referred to earlier. The SO control code corresponds to ASCII code 14.

```
10 LPRINT "This is the normal print mode."
20 LPRINT CHR$(14);
30 LPRINT "Expanded print mode"
40 END
```

The program output is:

```
This is the normal print mode.
Expanded print mode
```

The LPRINT command in line 20 sent an ASCII code 14 to the printer. The printer converted this code to a message with the meaning, "Turn on the expanded print mode." The expanded print mode was then activated.

The discussion to this point is fairly clearcut. However, only 15 characters in the ASCII table are available as control codes. The ESC (or escape) character is used to expand the list of available control codes. The message the ESC character carries is, "Treat the next character as a control code." Appendix B lists all the control codes.

As shown below, ESC corresponds to ASCII code 27.

```
10 LPRINT CHR$(27)"M";
20 LPRINT "Elite-sized characters mode"
30 LPRINT CHR$(27)"P";
40 LPRINT "Pica-sized characters mode"
50 END
```

The program output is:

```
Elite-sized characters mode
Pica-sized characters mode
```

In line 10, CHR$(27) is the ESC character. The next character sent to the printer is "M". Actually, ASCII code 77 is sent. But, we will let the computer and printer take care of the small details. The message "M" carries the following meaning when treated as a control code: "Change to the elite-sized character mode."

Similarly, the message "P" carries in line 30 is, "Cancel the elite-sized character mode." Notice that the "M" and the "P" are like all other control codes in that they were not printed.

PRINT BUFFER

The RX and FX Series printers both include a print buffer. A buffer is a temporary storage area used by the printer as a holding area for output data from the host computer.

The print buffer allows for quicker data transfer. With the buffer, the printer can accept a whole line of output data at once. Without the buffer, the printer could only accept one character at a time. That character would then be processed. The printer could then accept the next character from the host computer. Excessive time would be lost while the computer accomplished this "spoon feeding" of the printer.

Epson printers perform buffer-full print. When a full line of print data has been received, the next printable character causes the line to be output. A carriage return and a line feed will be executed after the line has been printed.

Control codes are never stored in the print buffer.* Control codes that cause a change in the print mode, a repositioning of the print head, or a paper feed force output of the buffer before they are executed. Other control codes will be executed when they are received.

The forced output of the buffer ensures that no data is lost, guarantees that spacing is performed as planned, and allows the printing of different modes on the same line.

The following program illustrates how an Epson printer uses the print buffer:

* This discussion of the print buffer differs from the explanation provided in the FX Series manuals. Although the scheme presented in either of the discussions would result in the same final output, the material provided in this book explains how the printer actually functions.

```
10 LPRINT "SOMETHING IN BUFFER ";
20 STOP
30 LPRINT CHR$(27)"M";
40 STOP
50 LPRINT "AND ANOTHER ";
60 STOP
70 LPRINT CHR$(27)"P";
80 STOP
90 LPRINT "YET ANOTHER"
100 STOP
110 LPRINT "THAT IS ALL"
120 END
```

When this program is run, the screen should display, "Break in 20". Notice that the data in line 10 has not been output yet. The print data is stored in the print buffer. Type CONT and press the Return key. The screen should display, "Break in 40". Printer output should read:

```
SOMETHING IN BUFFER
```

The print buffer was output because the print mode was changed in line 30.

Type CONT and press the Return key. The screen should display, "Break in 60". Notice that the data in line 50 is stored in the buffer.

Type CONT and press the Return key. The screen should display, "Break in 80". Also, the printer output should now read:

```
SOMETHING IN BUFFER AND ANOTHER
```

Notice that the added output,

```
AND ANOTHER
```

is printed on the same line, but in a different character size. The buffer was output because the print mode was changed in line 70.

Type CONT and press the Return key. The screen should display, "Break in 100". Also, the printer output should now read:

```
SOMETHING IN BUFFER AND ANOTHER YET ANOTHER
```

Notice that the added output,

```
YET ANOTHER
```

is printed on the same line, but in the original character size. Since the LPRINT command in line 90 does not end in a semicolon, a carriage return and a line feed were automatically added at the end of the print data. The carriage return caused the buffer to be output.

Type CONT and press the Return key. The printer output should now read:

```
SOMETHING IN BUFFER AND ANOTHER YET ANOTHER
THAT IS ALL
```

Notice that the added output,

```
THAT IS ALL
```

is printed on a new line. The new line was caused by the carriage return and line feed added to the data in line 90. Because the LPRINT command in line 110 does not end with a semicolon, the buffer containing "THAT IS ALL" was output. The next data would be printed beginning at the start of the next line.

RESETTING A MODE FROM BASIC

The Master Reset code allows the printer to be returned to the same state as when the power was first turned ON. That is, the buffer will be cleared, and all modes will be reset to their default conditions. The default conditions will be discussed in chapter 13.

The BASIC expression for the Master Reset code is:

CHR$(27)"@";

```
10 LPRINT CHR$(27)"M";
20 LPRINT "THIS IS IN ELITE"
30 LPRINT "THIS WILL BE LOST";
40 LPRINT CHR$(27)"@";
50 LPRINT "RETURN TO PICA WITH";
60 LPRINT "THE MASTER RESET"
70 END
```

The output from this program is:

```
THIS IS IN ELITE
RETURN TO PICA WITH THE MASTER RESET
```

Notice that the data in line 30 does not appear. The data was lost when the print buffer was cleared by the Master Reset code in line 40. Insertion of a blank LPRINT immediately before the Master Reset code will ensure that no data is lost.

CANcel and DELete

The CANcel code is available only to FX Series users. The CANcel code is a less powerful version of the Master Reset code. The CANcel code allows you to clear the buffer. This code does not affect printer modes.

The BASIC expression for the CAN code is:

CHR$(24);

```
10 LPRINT "SOMETHING IN THE BUFFER";
20 LPRINT CHR$(24);
30 LPRINT "ANOTHER IN THE BUFFER";
40 LPRINT
50 END
```

The program output is:

```
ANOTHER IN THE BUFFER
```

Notice that the data sent to the printer by the command in line 10 was never output. This data was erased from the buffer. The CAN code sent to the printer by the statement in line 20 caused the erasure.

The DELete code removes the previous text character from the buffer. This code does not affect control codes.

The BASIC expression for DEL is:

CHR$(127);

```
10 LPRINT "I will see you Monday and";
20 LPRINT CHR$(127)CHR$(127)CHR$(127);
30 LPRINT "or Tuesday."
40 END
```

The output from the program is:

```
I will see you Monday or Tuesday.
```

Notice that the three successive DELete codes removed the three letters "and".

The FX and RX Series printers also provide control codes to turn off each specific mode. These codes allow you to choose exactly which modes to deactivate. The control code to reset a specific mode will be presented immediately following the discussion of how that mode is set.

5

Pitch Modes

Introduction

In this chapter, we will discuss the pitch modes available on the FX and RX Series printers. The pitch mode defines the width of each character.

Both Series have three basic pitch modes and an expanded mode. Combining the expanded mode with the other three modes yields a total of six different character widths.

Character width defines the output's print style. Each of the pitch modes has its own distinctive print style.

Pica, Elite, and Compressed Modes

These three pitch modes are mutually exclusive. That is, only one can be used at a time. The pica mode has 10 characters per inch (cpi). The elite

mode has 12 cpi. These two modes are comparable to the standard character widths on a typewriter. The third pitch mode, compressed, has 17.16 cpi. The three modes have an order of precedence that will be detailed later in the chapter.

The pica mode is the factory set default condition for the pitch mode. This mode is considered the normal pitch mode. The output will be generated in the pica mode unless a different pitch mode is active.

```
10 LPRINT "This is the pica mode.";
20 LPRINT "It has 10 cpi."
30 END
```

The output from the program is:

```
This is the pica mode.It has 10 cpi.
```

Notice that no control codes were necessary to output the pica mode.

The control code to turn on the elite mode is ESC M. The elite mode is turned off by ESC P.

The BASIC expression for ESC M is:

CHR$(27)"M";

For ESC P, the expression is:

CHR$(27)"P";

```
10 LPRINT "This is the pica mode. ";
15 LPRINT "It has 10 cpi."
20 LPRINT CHR$(27)"M";
30 LPRINT "This is the elite mode. ";
35 LPRINT "It has 12 cpi."
40 LPRINT CHR$(27)"P";
50 LPRINT "Pica again.  Compare the";
55 LPRINT " two pitch modes."
60 END
```

The program output is:

```
This is the pica mode. It has 10 cpi.
This is the elite mode. It has 12 cpi.
Pica again.  Compare the two pitch modes.
```

Notice that the statement in line 20 turns the elite mode on. The statement in line 40 turns the elite mode off.

The control code used to turn on the compressed mode is SI. The compressed mode is turned off by DC2.

The BASIC expression for SI is:

CHR$(15);

For DC2, the expression is:

CHR$(18);

```
10 LPRINT "This is the pica mode. ";
15 LPRINT "It has 10 cpi."
20 LPRINT CHR$(27)"M";
30 LPRINT "This is the elite mode. ";
35 LPRINT "It has 12 cpi."
40 LPRINT CHR$(27)"P";
50 LPRINT CHR$(15);
60 LPRINT "This is the compressed mode. ";
65 LPRINT "It has 17.61 cpi."
70 LPRINT CHR$(18);
80 LPRINT "Pica again.  Compare the three";
85 LPRINT " pitch modes."
90 END
```

The program output is:

```
This is the pica mode. It has 10 cpi.
This is the elite mode. It has 12 cpi.
This is the compressed mode. It has 17.61 cpi.
Pica again.  Compare the three pitch modes.
```

Notice that the statement in line 50 activates the compressed mode. The statement in line 70 turns the compressed mode off.

Pitch Mode Precedence

As mentioned earlier, data will be output in the pica mode only if no other pitch mode is turned on. That means both the elite and compressed modes have precedence over the pica mode.

But, what happens if both the elite and compressed modes are turned on? The elite mode has precedence over the compressed mode. So, if both the elite and compressed modes are active, output will be generated in the elite mode. The compressed mode will be used if it is still active when the elite mode is turned off. The following program illustrates mode precedence:

```
10 LPRINT CHR$(15);
20 LPRINT "COMPRESSED MODE ACTIVE"
30 LPRINT CHR$(27)"M";
40 LPRINT "ELITE TAKES PRECEDENCE";
45 LPRINT " OVER COMPRESSED"
50 LPRINT CHR$(27)"P";
60 LPRINT "ELITE MODE OFF. COMPRESSED";
65 LPRINT " MODE ACTIVE"
70 LPRINT CHR$(18);
80 LPRINT "COMPRESSED MODE OFF.";
85 LPRINT " PICA NOW USED."
90 END
```

The program output is:

```
COMPRESSED MODE ACTIVE
ELITE TAKES PRECEDENCE OVER COMPRESSED
ELITE MODE OFF. COMPRESSED MODE ACTIVE
COMPRESSED MODE OFF. PICA NOW USED.
```

The statement in line 10 activates the compressed mode. When the elite mode is enabled in line 30, output is generated in elite characters. Deactivating the elite mode in line 50 leaves the compressed mode in precedence once again. Finally, turning the compressed mode off in line 70 returns the pitch mode to pica.

Enlarged Mode

The fourth pitch mode available is the enlarged mode. The enlarged mode can also be referred to as the expanded mode. The enlarged mode is used with whichever pitch mode presently has precedence. The enlarged mode defines the pitch to be twice the usual amount. Therefore, enlarged pica has 5 cpi, expanded elite has 6 cpi, and enlarged compressed has 8.58 cpi. The enlarged versions of the pica, elite, and compressed modes have the same precedence as the non-expanded versions discussed in the last section.

There are two different ways to turn on the enlarged mode. One of these consists of using the SO code to enable the expanded mode for only one line. This version of the enlarged mode is handy for printing column headings, titles and the like. The SO code is turned off by a line feed or by the DC4 code.

The BASIC expression for SO is:

CHR$(14);

For DC4, the expression is:

CHR$(20);

```
10 LPRINT CHR$(14);
20 LPRINT "     SOME TITLE"
30 FOR I=1 TO 5
40 LPRINT "TEXT TEXT TEXT TEXT TEXT TEXT TEXT"
50 NEXT I
60 LPRINT "TEXT TEXT ";
70 LPRINT CHR$(14);
80 LPRINT "TEXT ";
90 LPRINT CHR$(20);
100 LPRINT "TEXT TEXT TEXT"
110 END
```

The program output is:

```
              SOME  TITLE
TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT TEXT TEXT TEXT TEXT
TEXT TEXT TEXT  TEXT TEXT TEXT
```

The preceding program shows two possible uses of the enlarged mode. The first usage of the expanded mode is as a title. The second use of the enlarged print mode lends emphasis to a portion of the text.

Notice how the enlarged mode was turned off each time. In the title line example, the enlarged mode was deactivated by the line feed. In the text emphasis example, the expanded mode was turned off by DC4.

The second way to activate the enlarged mode is with the ESC W code. This enabling of the expanded mode remains on until it is deactivated.

The BASIC expression to enable the expanded print mode using ESC W is:

CHR$(27)"W1";

The expression to deactivate the enlarged print mode using ESC W is:

CHR$(27)"W0";

```
10 LPRINT "PICA MODE"
20 LPRINT CHR$(15);
30 LPRINT "COMPRESSED MODE"
40 LPRINT CHR$(27)"M";
50 LPRINT "ELITE MODE"
60 LPRINT CHR$(27)"W1";
70 LPRINT "ELITE AND EXPANDED MODES"
80 LPRINT CHR$(27)"P";
90 LPRINT "COMPRESSED AND EXPANDED MODES"
100 LPRINT CHR$(18);
110 LPRINT "PICA AND EXPANDED MODES"
120 LPRINT CHR$(27)"W0";
130 LPRINT "BACK TO PICA MODE"
140 END
```

The program output is:

```
PICA MODE
COMPRESSED MODE
ELITE MODE
ELITE AND EXPANDED MODES
COMPRESSED AND EXPANDED MODES
PICA AND EXPANDED MODES
BACK TO PICA MODE
```

Notice that the mode precedence still holds when the expanded mode is turned on. Also, compare the six available print pitch modes.

6

Weight Modes

Introduction

In the last chapter, we changed the print pitch in order to enable different print styles. In this chapter, we will change the print weight so as to enable improved print quality.

Two factors are involved in changing the weight mode. The first factor is print density. Print density is the number of dots used to form each character. The denser the print, the better its appearance. The second factor is the spacing between characters.

All of the weight modes decrease the print speed by one-half. This decreased print speed is 50 cps for the RX Series and 80 cps for the FX Series. Also, the increased density causes the ribbon to wear more quickly. So, increased print quality results in longer output time and shortened ribbon life.

In the last two sections of the chapter, we will discuss pitch and weight mode compatibility and precedence. Understanding the concepts in these sections is crucial when selecting mixed pitch and weight modes.

Double-Strike Mode

The double-strike mode affects the print density. When the double-strike mode is active, each line will be printed twice. The first pass is printed in the normal fashion. Figure 6.1 shows how two typical letters would appear after the first pass. An incremental line feed is then performed. The second pass is also printed normally. Notice that the incremental line feed causes the two printings of the line to be vertically offset by a slight amount. This vertical offset causes the dots in the characters of the second printing to fill in the vertical gaps remaining between the dots in the characters after the first printing. Figure 6.2 shows the appearance of the two typical letters upon completion of the second printing.

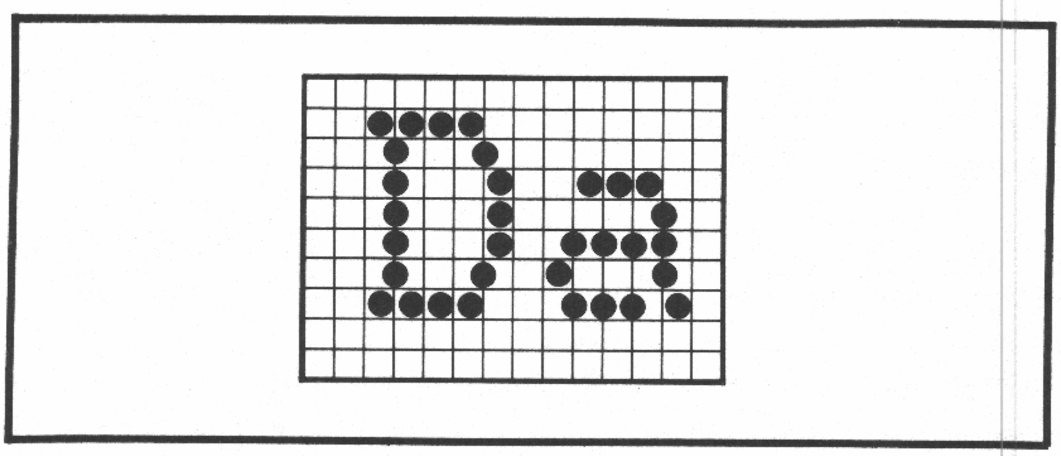


FIGURE 6.1. Two typical letters in single-strike

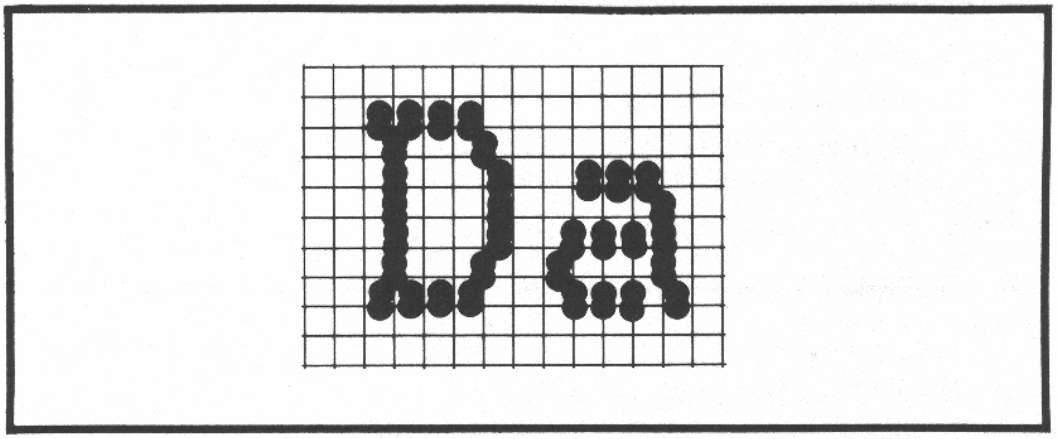


FIGURE 6.2. Two typical letters in double-strike

The double-strike mode is enabled by the ESC G code. The ESC H code is used to turn the double-strike mode off. ESC G can be represented in BASIC as follows using the CHR$ function:

CHR$(27)"G";

ESC H can be represented as:

CHR$(27)"H";

The use of ESC G and ESC H in BASIC is illustrated in the following example program:

```
10 LPRINT "NORMAL, SINGLE-STRIKE MODE."
20 LPRINT CHR$(27)"G";
30 LPRINT "DOUBLE-STRIKE MODE. IT IS DARKER."
40 LPRINT CHR$(27)"H";
50 LPRINT "BACK TO SINGLE-STRIKE."
60 END
```

The program output is:

```
NORMAL, SINGLE-STRIKE MODE.
DOUBLE-STRIKE MODE. IT IS DARKER.
BACK TO SINGLE-STRIKE.
```

Examine the output closely. Compare a "T" in the double-strike mode with a "T" in the single-strike mode. Notice that the vertical line of a "T" printed in the double-strike mode appears to be a solid line. Now, look at the vertical line of a "T" printed in the single-strike mode. This vertical line appears to be a column of dots. The command in line 20 enabled the double-strike mode. The statement in line 40 deactivated the double-strike mode.

Emphasized Mode

The emphasized mode affects the print density. When the emphasized mode is active, each dot in the character will be printed twice. The second dot is horizontally spaced so that it overlaps half of the first dot. This process fills in the horizontal gaps between adjacent dots in the character. Figure 6.3 shows two typical non-emphasized characters.

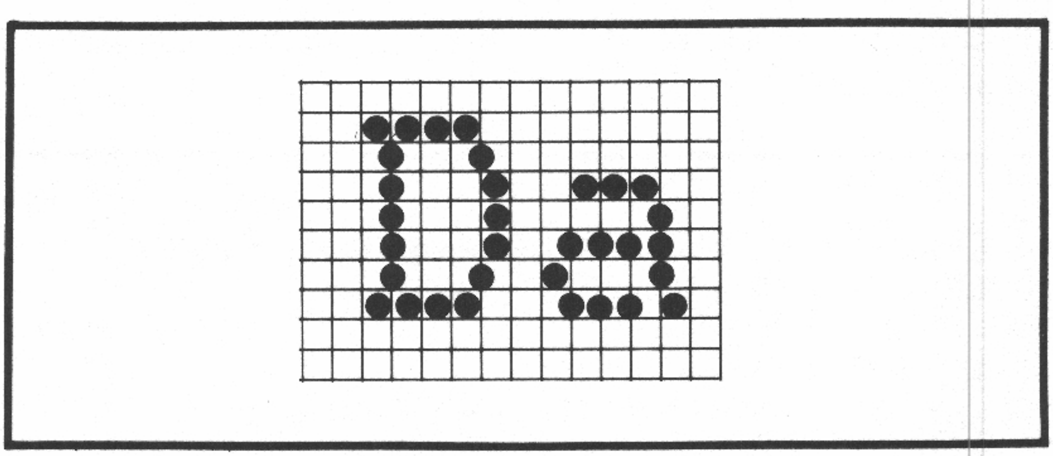


FIGURE 6.3. Two typical non-emphasized characters

Figure 6.4 shows how the same two characters look when they are emphasized. Notice how the second set of dots fills in the horizontal gaps in the characters.

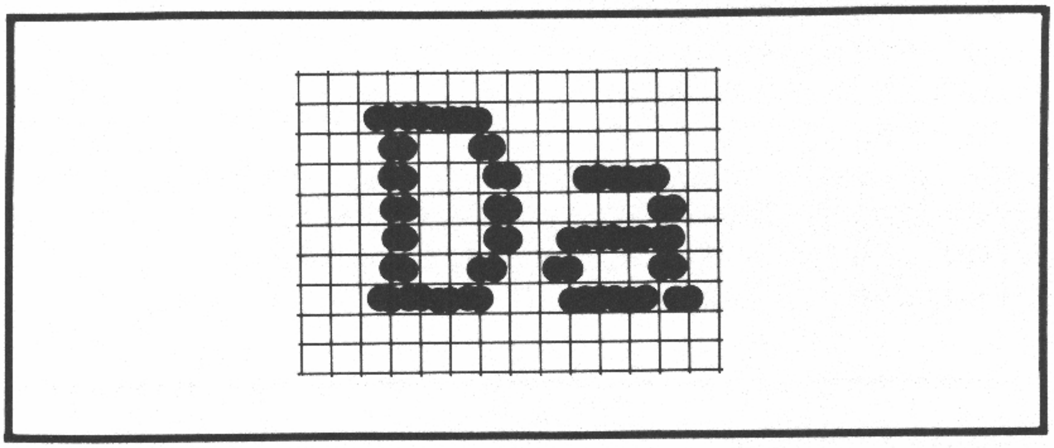


FIGURE 6.4. Two typical emphasized characters

The emphasized mode is enabled by the ESC E code. The ESC F code is used to turn the emphasized mode off. ESC E can be represented in BASIC as follows:

CHR$(27)"E";

ESC F can be represented as:

CHR$(27)"F";

The use of ESC E and ESC F in BASIC is illustrated in the following program:

```
10 LPRINT "SINGLE-STRIKE"
20 LPRINT CHR$(27)"G";
30 LPRINT "DOUBLE-STRIKE"
40 LPRINT CHR$(27)"H";
50 LPRINT CHR$(27)"E";
60 LPRINT "EMPHASIZED"
70 LPRINT CHR$(27)"G";
80 LPRINT "DOUBLE-STRIKE AND EMPHASIZED"
90 LPRINT CHR$(27)"F"CHR$(27)"H";
100 LPRINT "SINGLE-STRIKE AGAIN"
110 END
```

The program output is:

```
SINGLE-STRIKE
DOUBLE-STRIKE
EMPHASIZED
DOUBLE-STRIKE AND EMPHASIZED
SINGLE-STRIKE AGAIN
```

Examine the output closely. Compare an "I" in the emphasized mode with an "I" in the single-strike mode. Notice that the horizontal line of an "I" printed in the emphasized mode appears to be a solid line. Now look at the horizontal line of an "I" printed in the single-strike mode. This horizontal line appears to be a row of dots. The command in line 50 enabled the emphasized mode. The statement in line 90 turned the emphasized and double-strike modes off.

The command in line 90 could have been replaced with a simpler expression:

LPRINT CHR$(27)"@";

This command uses the Master Reset Code to reset the print modes to their default settings.

Notice from the fourth line of output that the emphasized and double-strike modes can be combined. This combination is the closest the Epson printer can come to imitating typed output.

Proportional Mode (FX Printers Only)

When the proportional mode is active, all characters are printed with a uniform amount of space between them. Epson printers usually generate output using identical widths for each character. However, a character such as "W" is much wider than a character such as "i". Figure 6.5 illustrates this difference in width. Regardless of which character is being generated, the printer will use a width large enough to accommodate a "W". Therefore, when an "i" or other narrow character is generated, extra blank space will be output on either side of the character. The proportional mode will strip this extra blank space away. Figure 6.6 shows identical lines of print generated in both non-proportional and proportional modes. All modes other than the proportional mode are non-proportional modes. Notice that the proportional mode makes little difference for the wide characters on the left. But, for the narrow characters on the right, the proportional mode changes the spacing a great deal. The characters generated by the proportional mode are darker than the other characters, because all output created by the proportional mode is automatically emphasized.

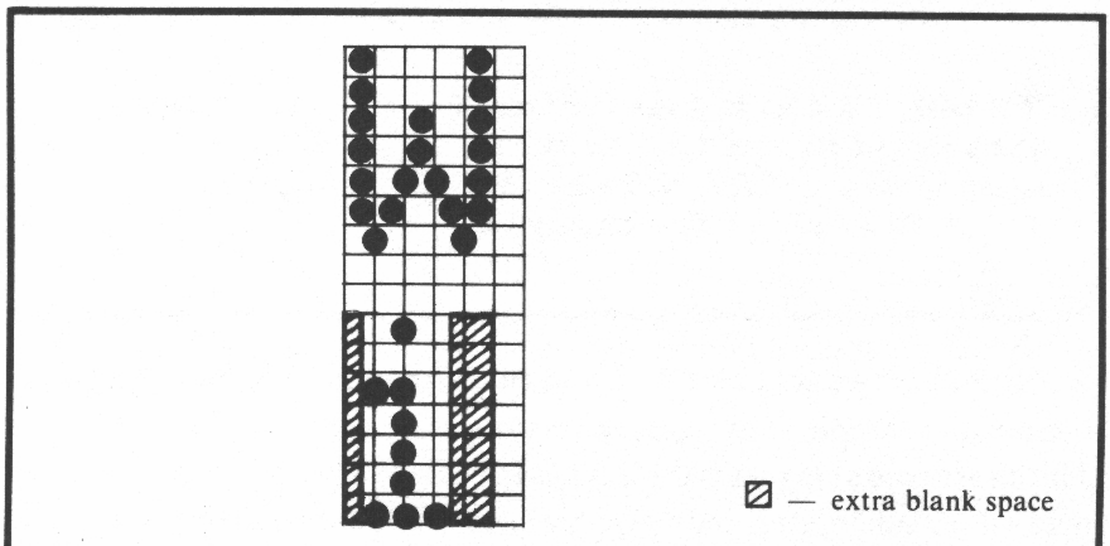


FIGURE 6.5. The difference between two characters

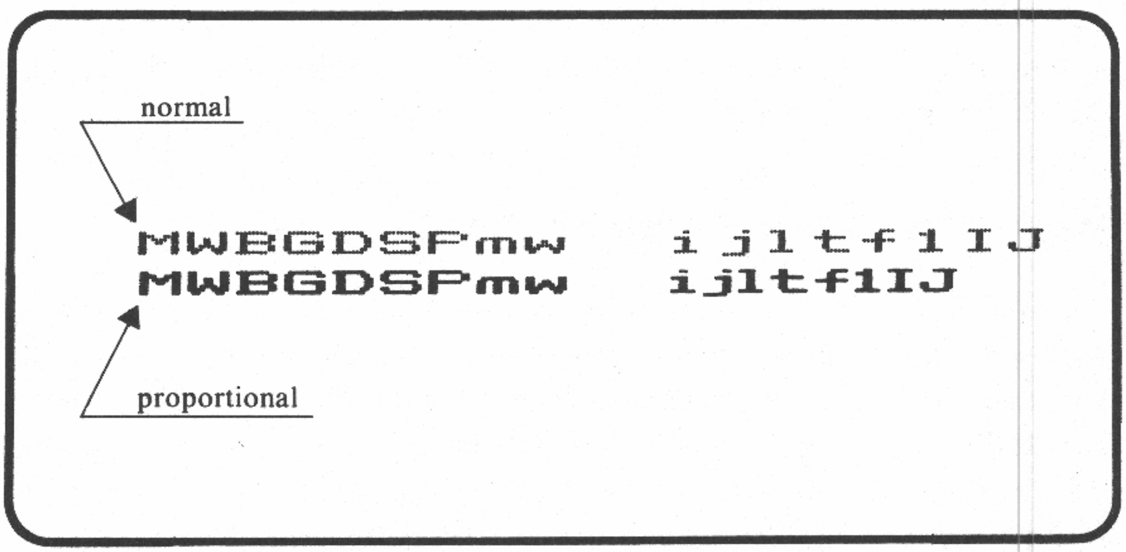


FIGURE 6.6. The difference between the normal and proportional modes

Consider the following output:

```
The little red engine
thinks it can.   It
says, 'I think I can.
I think I can.'

The little red engine
thinks it can.  It
says, 'I think I can.
I think I can.'
```

Which of the two texts did you find easier to read? You probably chose the bottom sample. This sample was printed in the proportional mode. This more readable text is the advantage that the proportional mode has over all the other pitch and weight modes available on the Epson printer.

Since different characters in the proportional mode have different widths, the proportional mode characters do not line up in vertical

columns. Therefore, the proportional mode should not be used to generate text where characters must line up vertically, such as tables. Any other combination of pitch and weight modes should be used to generate such text as tables.

The proportional mode can use only the pica or enlarged pica pitch modes to generate output. Also, the proportional mode is only available on FX Series printers.

The proportional mode is turned on by sending the ESC p1 code. The proportional mode is deactivated by sending the ESC p0 code. ESC p1 can be represented as follows in BASIC:

CHR$(27)"p1";

ESC p0 can be represented as:

CHR$(27)"p0";

The use of ESC p1 and ESC p0 in BASIC is illustrated in the following program:

```
10 LPRINT CHR$(27)"p1";
20 LPRINT "PROPORTIONAL:"
30 LPRINT "This proportional mode example"
40 LPRINT "is done in regular pica instead of"
50 LPRINT "expanded pica like the previous"
60 LPRINT "examples in the text."
70 LPRINT CHR$(27)"p0"
80 LPRINT "NOT PROPORTIONAL:"
90 LPRINT "This example of the proportional
          mode"
100 LPRINT "is done in regular pica instead
           of"
110 LPRINT "expanded pica like the previous"
120 LPRINT "examples in the text."
130 END
```

The program output is:

```
PROPORTIONAL:
This proportional mode example
is done in regular pica instead of
expanded pica like the previous
examples in the text.

NOT PROPORTIONAL:
This example of the proportional mode
is done in regular pica instead of
expanded pica like the previous
examples in the text.
```

Notice the differences between the proportional and non-proportional outputs. The proportional mode was activated by the command in line 10. The proportional mode was turned off by the statement in line 70.

Weight Mode Precedence

In chapter 5, we saw that the pitch modes had a set order of precedence. For example, if both the elite and the compressed modes were active, the elite mode would take precedence, and all output would be generated using the elite mode. The weight modes also have a set order of precedence.

The single-strike mode is the default weight mode. If none of the other weight modes are active, output will be generated using the single-strike mode. Figure 6.7 illustrates the precedence of the weight modes. Notice that the single-strike mode is at the bottom. This placement indicates that the single-strike mode has the lowest precedence.

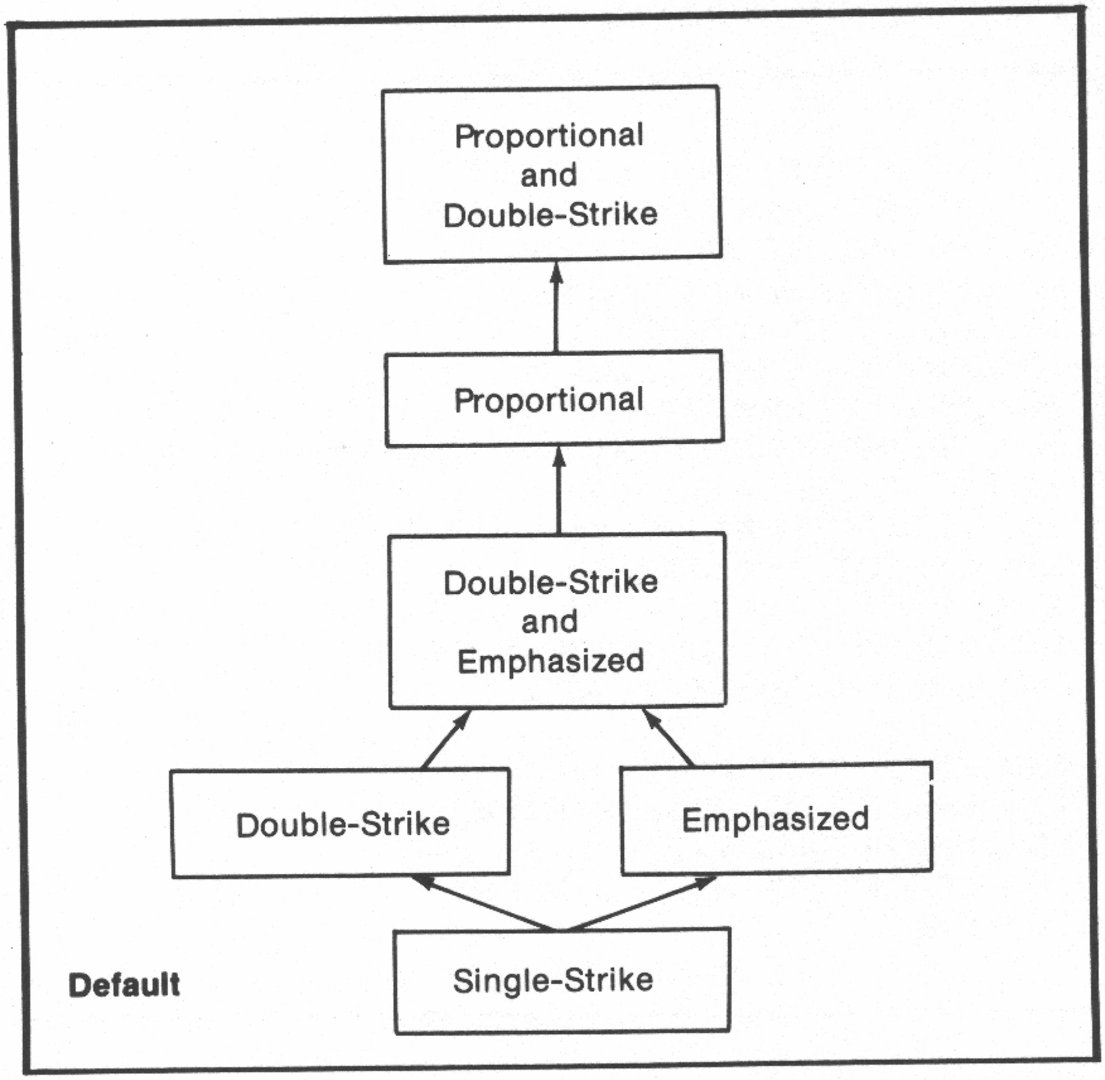


FIGURE 6.7. Weight mode precedence

The precedence of the other weight modes can be identified from figure 6.7. Notice that the double-strike and emphasized modes have equal precedence. The combination of both the double-strike and emphasized modes takes precedence over either of those single modes. The proportional mode occupies the next level of precedence. Finally, the combination of the proportional and double-strike modes takes the highest precedence.*

* **Note:** the Epson FX Series owner's manuals indicate that the double-strike mode will not function when the proportional mode is active. This writer found that the two modes were compatible.

The following program illustrates weight mode precedence:

```
10 LPRINT "SINGLE-STRIKE"
20 LPRINT CHR$(27)"G";
30 LPRINT "DOUBLE-STRIKE"
40 LPRINT CHR$(27)"H";
50 LPRINT CHR$(27)"E";
60 LPRINT "EMPHASIZED"
70 LPRINT CHR$(27)"G";
80 LPRINT "DOUBLE-STRIKE AND EMPHASIZED"
90 LPRINT CHR$(27)"H";
100 LPRINT CHR$(27)"p1";
110 LPRINT "PROPORTIONAL"
120 LPRINT CHR$(27)"F";
130 LPRINT "STILL PROPORTIONAL"
140 LPRINT CHR$(27)"G";
150 LPRINT "PROPORTIONAL AND DOUBLE-STRIKE"
160 LPRINT CHR$(27)"p0";
170 LPRINT "DOUBLE-STRIKE"
180 LPRINT CHR$(27)"H";
190 LPRINT "SINGLE-STRIKE"
200 END
```

The program output is:

```
SINGLE-STRIKE
DOUBLE-STRIKE
EMPHASIZED
DOUBLE-STRIKE AND EMPHASIZED
PROPORTIONAL
STILL PROPORTIONAL
PROPORTIONAL AND DOUBLE-STRIKE
DOUBLE-STRIKE
SINGLE-STRIKE
```

The command in line 10 caused the first line of output. This output was generated using the single-strike mode, because no other weight modes were active. The statement in line 20 enabled the double-strike mode. The command in line 30 caused the second line of output. This output was generated using the double-strike mode.

The statements in lines 40 and 50 turned off the double-strike mode and activated the emphasized mode, respectively. The command in line 60 caused the third line of output. This output was generated using the emphasized mode.

The statement in line 70 enabled the double-strike mode. The command in line 80 caused the fourth line of output. This output was generated using both the double-strike and emphasized modes.

The statements in lines 90 and 100 deactivated the double-strike mode and enabled the proportional mode, respectively. The command in line 110 caused the fifth line of output. This output was generated using the proportional mode.

The statement in line 120 turned the emphasized mode off. The command in line 130 caused the sixth line of output. This output was generated using the proportional mode. Notice that the output was still emphasized. The proportional mode automatically emphasizes all output. Therefore, the proportional mode ignores the current setting of the emphasized mode.

The statement in line 140 turned the double-strike mode on. The command in line 150 caused the seventh line of output. This output was generated using the proportional and double-strike modes.

The statement in line 160 deactivated the proportional mode. The command in line 170 caused the eighth line of output. This output was generated using the double-strike mode. Notice that the statement in line 120 did actually turn the emphasized mode off, although this change was not observable until line 170 was executed.

The statement in line 180 turned the double-strike mode off. The command in line 190 caused the last line of output. This output was generated using the single-strike mode.

Mixing Weight and Pitch Modes

The example programs used so far in this chapter to illustrate the weight modes were combined only with the pica pitch mode. The weight modes can be combined with other pitch modes as well. However, not all of the combinations of weight and pitch can function together. The combinations that cannot function together are called incompatible combinations. There is an established order of precedence between the modes contained in any incompatible combination. For example, the elite and emphasized modes are incompatible. When both are activated, the output will be generated using the elite mode. The emphasized mode is not used. Therefore, the elite mode has precedence over the emphasized mode.

Table 6.1 lists the result of combining the different pitch and weight modes. Wherever a check mark appears in a box, the modes of the top of the column and the left of the row are compatible and may be used together. If the modes at the top of the column and the left of the row are not compatible, the box will contain a message telling you which mode takes precedence and the style of output generated by the combination. For example, examine the box at the intersection of the column headed by "Emphasized" and the row labeled "Compressed". The messages in the box indicate that the emphasized mode will take precedence over the compressed mode, and that output will be generated using the emphasized and pica modes. Notice that the compressed mode does not function even though it is active.

The following program illustrates one example of pitch and weight mode precedence:

```
10 LPRINT CHR$(27)"M";
20 LPRINT CHR$(27)"E";
30 LPRINT CHR$(27)"G";
40 LPRINT "ELITE HAS PRECEDENCE";
45 LPRINT " OVER EMPHASIZED."
50 LPRINT "OUTPUT IS IN DOUBLE-STRIKE ELITE."
60 END
```

The program output is:

```
ELITE HAS PRECEDENCE OVER EMPHASIZED.
OUTPUT IS IN DOUBLE-STRIKE ELITE.
```

The command in line 10 activates the elite mode. The statement in line 20 turns on the emphasized mode. The command in line 30 enables the double-strike mode. We have purposefully activated an incompatible combination of pitch and weight modes. Table 6.1 indicates that the output from the combination of the elite, emphasized, and double-strike modes should be generated using the elite and double-strike modes only. Check the output. It is elite and double-strike. It is not emphasized.

Table 6.1. Mode compatibility and precedence

PITCH MODE	WEIGHT MODE: Single-Strike	Double-Strike	Emphasized	Double-Strike Emphasized	Proportional	Double-Strike Proportional
Pica	✓	✓	✓	✓	✓	✓
Enlarged Pica	✓	✓	✓	✓	✓	✓
Elite	✓	✓	Precedence: Elite Output: Single-strike elite	Precedence: Elite Output: Double-strike elite	Precedence: Elite Output: Single-strike elite	Precedence: Elite Output: Double-strike elite
Enlarged Elite	✓	✓	Precedence: Elite Output: Single-strike enlarged elite	Precedence: Elite Output: Double-strike enlarged elite	Precedence: Elite Output: Single-strike enlarged elite	Precedence: Elite Output: Double-strike enlarged elite
Compressed	✓	✓	Precedence: Emphasized Output: Emphasized pica	Precedence: Emphasized Output: Emphasized double-strike pica	Precedence: Proportional Output: Single-strike proportional pica	Precedence: Proportional Output: Double-strike proportional pica
Enlarged Compressed	✓	✓	Precedence: Emphasized Output: Emphasized, enlarged pica	Precedence: Emphasized Output: Emphasized, double-strike enlarged pica	Precedence: Proportional Output: Single-strike, enlarged proportional pica	Precedence: Proportional Output: Double-strike, enlarged proportional pica

7

Master Select Mode (FX Series Only)

Introduction

The Master Select Mode is available only on the FX Series printers. This mode adds no new features to the printer. Instead, the Master Select Mode allows you to choose any compatible combination of pitch and weight modes. Pitch and weight mode compatibility was discussed in the last chapter.

The advantage of the Master Select Mode lies in the fact that you can replace several program statements with a single command. Furthermore, the Master Select Mode will not allow you to accidentally enable incompatible modes. Using the Master Select Mode is a quick and sure way to choose the style of printer output.

Using the Master Select Mode

Use of the Master Select Mode causes the printer to deactivate all presently active pitch and weight modes. Following this purge, the specified pitch and weight modes are enabled.

The Master Select Mode is represented by the ESC ! code. In BASIC, the ESC ! code can be expressed in two different forms:

CHR$(27)"!*b*"   or   CHR$(27)"!"CHR$(*n*)

where *b* represents a character from table 7.1, and *n* represents a number from table 7.1.

Table 7.1 identifies the values of *b* and *n* to specify for any compatible combination of pitch and weight. An "N/A" in table 7.1 denotes that the specified combination of pitch and weight is incompatible and cannot be selected using the Master Select Mode.

Table 7.1. Characters and numbers used in the code for the Master Select Mode

PITCH MODE	WEIGHT MODE								
	Single-strike		Double-strike		Emphasized		Double-strike Emphasized		Proportional
	CHAR	NUMB	CHAR	NUMB	CHAR	NUMB	CHAR	NUMB	
Pica	@	0	P	16	H	8	X	24	N/A
Enlarged Pica	b̸†	32	*	40	0††	48	8	56	N/A
Elite	A	1	Q	17	N/A		N/A		N/A
Enlarged Elite	!	33	1	49	N/A		N/A		N/A
Compressed	D	4	T	20	N/A		N/A		N/A
Enlarged Compressed	$	36	4	52	N/A		N/A		N/A

† b̸ stands for a blank space (i.e. CHR$(27)"! ").

†† This is a zero character.

There is one exception to the rule that all compatible pitch and weight modes can be specified. The exception is that no combination using the proportional mode can be selected using the Master Select Mode.

The three following programs all generate identical output. We will use these three programs to illustrate the advantages of using the Master Select Mode and to show how to utilize the two different forms of the ESC ! code.

The first example program uses only the single mode codes discussed in chapters 5 and 6:

```
10 LPRINT CHR$(27)"W1";
20 LPRINT CHR$(27)"G";
30 LPRINT CHR$(27)"E";
40 LPRINT "DOUBLE-STRIKE AND"
50 LPRINT "EMPHASIZED ENLARGED"
60 LPRINT "PICA."
70 LPRINT CHR$(27)"W0";
80 LPRINT CHR$(27)"F";
90 LPRINT CHR$(27)"M";
100 LPRINT "DOUBLE-STRIKE ELITE."
110 LPRINT CHR$(27)"P";
120 LPRINT CHR$(27)"H";
130 LPRINT "SINGLE-STRIKE PICA."
140 END
```

The program output is:

```
DOUBLE-STRIKE AND
EMPHASIZED ENLARGED
PICA.
DOUBLE-STRIKE ELITE.
SINGLE-STRIKE PICA.
```

The commands in lines 10 through 30 enabled the enlarged, double-strike and emphasized modes. The statements in lines 70 through 90 deactivated the enlarged and emphasized modes and turned the elite mode on. The commands in lines 110 and 120 cancelled the elite and double-strike modes.

The second example program shows how the same program would look using the Master Select Mode:

```
10 LPRINT CHR$(27)"!8";
20 LPRINT "DOUBLE-STRIKE AND"
30 LPRINT "EMPHASIZED ENLARGED"
40 LPRINT "PICA."
50 LPRINT CHR$(27)"!Q";
60 LPRINT "DOUBLE-STRIKE ELITE."
70 LPRINT CHR$(27)"!@";
80 LPRINT "SINGLE-STRIKE PICA."
90 END
```

The program output is:

```
DOUBLE-STRIKE AND
EMPHASIZED ENLARGED
PICA.
DOUBLE-STRIKE ELITE.
SINGLE-STRIKE PICA.
```

Notice that the program is five lines shorter than the first example program, but still generates the same output. The command in line 10 of the second example program replaced the statements in lines 10 through 30 of the first example program. Notice the *8* used in line 10 of the second example program. Look at table 7.1 and find the character *8*. Scan across the row to find that the character *8* represents the enlarged pica pitch mode. Scan up the column to find that the character *8* also represents the double-strike and emphasized weight modes. These are the same modes that the statements in lines 10 through 30 of the first example program enabled.

The command in line 50 of the second example program replaced the statements in lines 70 through 90 of the first example program. Notice the *Q* in line 50 of the second example program. You can see on table 7.1 that the character *Q* represents a combination of the elite and double-strike modes. Finally, the command in line 70 of the second example program replaced the statements in lines 110 and 120 of the second example program. Notice the @ in line 70 of the second example program. You can easily see on table 7.1 that the @ character represents the single-strike pica mode.

The last example program illustrates the technique of putting the Master Select Mode in a subroutine:

```
10 MODE=56:GOSUB 100
20 LPRINT "DOUBLE-STRIKE AND"
30 LPRINT "EMPHASIZED ENLARGED"
40 LPRINT "PICA."
50 MODE=17:GOSUB 100
60 LPRINT "DOUBLE-STRIKE ELITE."
70 MODE=0:GOSUB 100
80 LPRINT "SINGLE-STRIKE PICA."
90 END
100 LPRINT CHR$(27)"!"CHR$(MODE);
110 RETURN
```

The program output is:

```
DOUBLE-STRIKE AND
EMPHASIZED ENLARGED
PICA.
DOUBLE-STRIKE ELITE.
SINGLE-STRIKE PICA.
```

Notice that all changes in the print mode are now made in the subroutine. The form of the ESC ! code used specifies a number instead of a character, as in the preceding program. That number is specified as the numeric variable, MODE.

In line 10, MODE is set equal to 56 before the subroutine is called. Find the number 56 in table 7.1. It is located to the right of the character *8*. The number 56 specifies a combination of double-strike, emphasized and enlarged pica modes. The statements in line 50 and 70 operate in a similar fashion, except that each sets MODE equal to a different number.

8

Print Enhancement and Miscellaneous Modes

Introduction

In this chapter, we will discuss several modes that do not logically complement the material covered in other chapters.

The last section of this chapter will be of particular interest to users of computers that control only seven data lines on their parallel printer interfaces. Generally, eight data lines are controlled by the host computer.

Underlined Mode

When the underlined mode is active, all output will be generated with an underscore.

The ESC - code controls the underlined mode. The BASIC expression to activate the underlined mode is:

CHR$(27)"-1";

The expression to turn the underlined mode off is:

CHR$(27)"-0";

The following program illustrates the use of the ESC - code:

```
10 LPRINT CHR$(27)"-1";
20 LPRINT "UNDERLINED MODE"
30 LPRINT CHR$(27)"-0";
40 LPRINT "NORMAL AGAIN"
50 END
```

The program output is:

```
UNDERLINED MODE
NORMAL AGAIN
```

The command in line 10 enabled the underlined mode. The command in line 30 turned the underlined mode off.

Script Modes

When a script mode is turned on, all output will be generated on either the upper or lower half of the print line. Which half is used depends on which of the two script modes is active. The specifics of each of the two script modes are detailed later in this section.

Characters 6.3 hundredths of an inch (1.6mm) in height are used to generate output using the script mode. This character height is slightly greater than one-half the height of a non-script character. Enabling a script mode will not change the width of a character.

In order to form such small characters, the printer must use double-strike printing. The script modes do not change the setting of the double-strike mode. Only one script mode can be active at a time. Enabling one script mode will deactivate the other script mode if it is turned on.

SUPERSCRIPT MODE

When the superscript mode is enabled, all output will be generated on the top half of the print line using the smaller script characters.

The ESC S0 code enables the superscript mode. The ESC T code turns the superscript mode off.

The BASIC expression for ESC S0 is:

CHR$(27)"S0";

The expression for ESC T is:

CHR$(27)"T";

The following program illustrates the use of the superscript mode:

```
10 LPRINT CHR$(27)"S0";
20 LPRINT "SUPERSCRIPT MODE"
30 LPRINT CHR$(14);
40 LPRINT "EXPANDED SUPERSCRIPT"
50 LPRINT CHR$(27)"T";
60 LPRINT "BACK TO NORMAL"
70 END
```

The program output is:

```
SUPERSCRIPT MODE
E X P A N D E D   S U P E R S C R I P T
BACK TO NORMAL
```

The statement in line 10 enabled the superscript mode. The command in line 50 deactivated the superscript mode.

SUBSCRIPT MODE

When the subscript mode is enabled, all output will be generated on the bottom half of the print line using the smaller characters.

The ESC S1 code enables the subscript mode. The ESC T code turns the subscript mode off.

The BASIC expression for ESC S1 is:

CHR$(27)"S1";

The expression for ESC T is:

CHR$(27)"T";

The following program illustrates the use of the subscript mode:

```
10 LPRINT CHR$(27)"S1";
20 LPRINT "SUBSCRIPT MODE"
30 LPRINT CHR$(14);
40 LPRINT "EXPANDED SUBSCRIPT"
50 LPRINT CHR$(27)"T";
60 LPRINT "BACK TO NORMAL"
70 END
```

The program output is:

```
SUBSCRIPT MODE
E X P A N D E D   S U B S C R I P T
BACK TO NORMAL
```

The statement in line 10 enabled the subscript mode. The command in line 50 deactivated the subscript mode.

A PRACTICAL EXAMPLE -- SCRIPT MODES

In this section, we detail a practical use of the two script modes. The example shows how to output a mathematical expression using your printer:

```
10 LPRINT CHR$(27)"M";
20 LPRINT "A";
30 LPRINT CHR$(27)"S0";
40 LPRINT "2";
50 LPRINT CHR$(27)"T";
60 LPRINT "=";
70 FOR I=1 TO 3
80 LPRINT "(a";
90 LPRINT CHR$(27)"S1";
100 LPRINT CHR$(48+I);
110 LPRINT CHR$(27)"T)";
120 LPRINT CHR$(27)"S02";
130 LPRINT CHR$(27)"T+";
140 NEXT I
150 LPRINT CHR$(27)"S0...";
160 LPRINT CHR$(27)"T+(a";
170 LPRINT CHR$(27)"S1n";
180 LPRINT CHR$(27)"T)";
190 LPRINT CHR$(27)"S02"
200 LPRINT CHR$(27)"@"
210 END
```

The program output is:

$$A^2=(a_1)^2+(a_2)^2+(a_3)^2+\cdots+(a_n)^2$$

The superscript mode was used to generate the exponents. The subscript mode was used to generate the variable subscripts. The ellipsis in the output was generated by sending three periods to the printer while the superscript mode was enabled. All other symbols were generated

without using the script modes. Notice that all of the output was generated using the elite mode. The script modes are compatible with all of the pitch modes.

Take special note of line 110. In this line, the subscript mode was deactivated and the right parentheses was sent to the printer. Notice that the T and the) appear together inside the quote marks. This technique is perfectly acceptable and can be used to reduce the number of program statements. The method could have been used even more extensively. Consider the following program:

```
10 LPRINT CHR$(27)"MA"CHR$(27)"S02"CHR$(27)"T=
";
20 FOR I=1 TO 3
30 LPRINT "(a"CHR$(27)"S1"CHR$(48+I)CHR$(27)"T
)"CHR$(27)"S02"CHR$(27)"T+";
40 NEXT I
50 LPRINT CHR$(27)"S0..."CHR$(27)"T+(a"CHR$(27
)"S1n"CHR$(27)"T)"CHR$(27)"S02"CHR
$(27)"T"CHR$(27)"@"
60 END
```

The output from this program is the same as the output from the preceding program:

$$A^2=(a_1)^2+(a_2)^2+(a_3)^2+\cdots+(a_n)^2$$

While the second program is much harder to decipher, it accomplishes the same task with fewer statements.

Italic Mode

When the italic mode is active, all output will be generated using an italic character set. Normal output is generated using a Roman character set. The characters in the two sets are the same except for their appearance. Both character sets appear in the ASCII table in appendix A.

The italic character set can be selected using the ESC 4 code. The ESC 5 code returns the printer to the Roman character set.

The BASIC expression for ESC 4 is:

CHR$(27)"4";

For ESC 5, the expression is:

CHR$(27)"5";

The following program illustrates the use of the ESC 4 and ESC 5 codes:

```
10 LPRINT CHR$(27)"4";
20 LPRINT "THE ALTERNATE CHARACTER SET:"
30 GOSUB 80
40 LPRINT CHR$(27)"5"
50 LPRINT "THE ROMAN CHARACTER SET:"
60 GOSUB 80
70 END
80 FOR I=31 TO 126 STEP 32
90 FOR J=0 TO 31
100 LPRINT CHR$(I+J);
110 NEXT J
120 LPRINT
130 NEXT I
140 RETURN
```

The program output is:

```
THE ALTERNATE CHARACTER SET:
 !"#$%&'()*+,-./0123456789:;<=>
?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
_`abcdefghijklmnopqrstuvwxyz{|}~

THE ROMAN CHARACTER SET:
 !"#$%&'()*+,-./0123456789:;<=>
?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
_`abcdefghijklmnopqrstuvwxyz{|}~
```

The command in line 10 selected the italic character set. The statement in line 40 returned the printer to the Roman character set. The FOR statement in line 80 sets I to the ASCII code value corresponding to the beginning of the Roman character set. The step of 32 is set so that the values output by the inner loop are skipped in the outer loop.

Backspace

The backspace allows you to move the carriage one space to the left. After a backspace, the next character output by the printer will overstrike the character that was first printed in that space. The net result is that two characters are printed in the same space.

The BS code controls the backspace feature. Each occurrence of a BS code results in one backspace. The BASIC expression for BS is:

```
CHR$(8);
```

The following example program shows how you can print several mathematical symbols that are not available in the ASCII character set:

```
10 LPRINT "="CHR$(8)"/";
20 LPRINT "  ~"CHR$(8)"-";
30 LPRINT "  >"CHR$(8)"_";
40 LPRINT "  <"CHR$(8)"/";
50 LPRINT CHR$(27)"S0 +"CHR$(8)CHR$(27)"S1-"
60 LPRINT CHR$(27)"T"
70 END
```

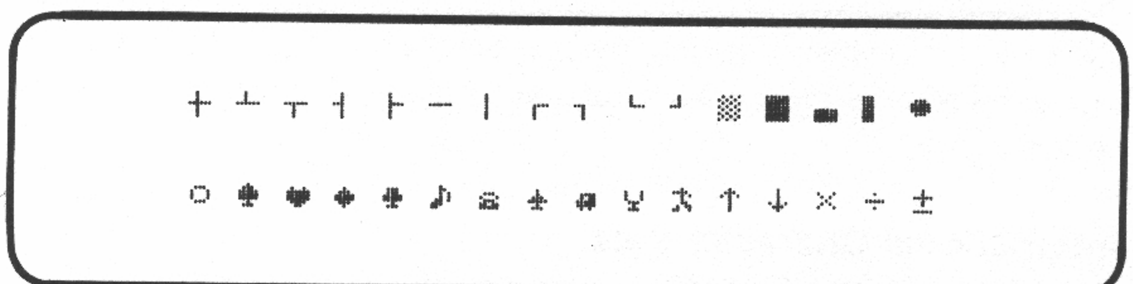
The program output is:

```
≠ ≃ ≥ ≮ ±
```

Notice that each LPRINT command created one of the symbols. Of special significance is the ± symbol. The two script modes were used along with the backspace to form the symbol. The LPRINT command in line 50 created the ± symbol. You should be able to generate almost any mathematical symbol you need by using the backspace feature. The CHR$(8) in each of the LPRINT commands caused the backspace.

Graphics Characters (RX Printers Only)

The graphics character set is a special set of symbols:



When the graphics character set is selected, the characters corresponding to ASCII codes 128 through 159 will be replaced by the graphics characters. The characters that are replaced are all control characters. All of the control characters have another corresponding ASCII code. So, all of the control codes remain accessible.

The ESC m code controls selection of the graphics character set. The BASIC expression for ESC m has the form:

CHR$(27)"m"CHR$(*n*);

where *n* represents a numerical value of either 0 or 4. If *n* is 0, the usual control characters will be selected. If *n* is 4, the graphic characters will be selected. The following program illustrates the use of the ESC *m* code:

```
10 LPRINT CHR$(27)"m"CHR$(4);
20 LPRINT "I had an A"CHR$(145)"A"CHR$(146);
30 LPRINT "A"CHR$(147)"9"CHR$(148)"9";
40 LPRINT CHR$(145)", FULLHOUSE!!!, to win"
50 LPRINT "the biggest pot of the night."
60 LPRINT CHR$(27)"m"CHR$(0);
70 END
```

The program output is:

```
I had an A♣A♠A♦9♣9♣, FULLHOUSE!!!, to win
the biggest pot of the night.
```

The statement in line 10 enables the graphics character set. The statement in line 60 returns the printer's version of the ASCII table to its usual composition.

International Character Sets

International character sets allow letters and symbols that are not used in the United States to be output. The RX Series has 11 international character sets. The FX Series has 9 sets. Table 8.1 shows these international character sets.

In the international character sets, a different letter or symbol always replaces a USA symbol. Note that the bulk of the character set is never

changed. The first row of symbols in table 8.1 shows the USA characters that may be different in an international character set.

Table 8.1. International character sets

COUNTRY	*b*												
USA	0	#	$	@	[	\	]	^	`	{	¦	}	~
FRANCE	1	#	$	à	°	ç	§	^	`	é	ù	è	¨
GERMANY	2	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
ENGLAND	3	£	$	@	[	\	]	^	`	{	¦	}	~
DENMARK I	4	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
SWEDEN	5	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
ITALY	6	#	$	@	°	\	é	^	ù	à	ò	è	ì
SPAIN	7	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
JAPAN	8	#	$	@	[	¥	]	^	`	{	¦	}	~
NORWAY(RX only)	9	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
DENMARK II(RX only)	10	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü

The ESC R code controls the selection of an international character set. The BASIC expression for ESC R has the form:

CHR$(27)"R"CHR$(*b*);

where *b* is a number or numerical expression. The value indicated for *b* specifies an international character set. The possible values of *b* are listed in table 8.1. The following program illustrates how to select and use an international character set:

```
10 LPRINT CHR$(27)"R"CHR$(7);
20 LPRINT "SPANISH--Ni|o"
30 LPRINT CHR$(27)"R"CHR$(0);
40 LPRINT "ENGLISH--Boy"
50 END
```

The program output is:

```
SPANISH--Niño
ENGLISH--Boy
```

The command in line 10 selected the international character set for Spain. Notice in line 20 that the ñ was sent to the printer as a ¦. Whenever you are using an international character set, you should use the USA character on your keyboard that corresponds to the desired international character.

Table 8.1 can be used to find the correct corresponding character. First, locate the desired international character in the table. Then, scan up the column to the row labelled USA. Use the character found in the USA row. The printer will automatically "translate' the USA character to the desired international character when the output is generated. The command in line 30 selected the USA character set.

Special Speed Modes

The FX Series printers have two special print speed modes. Only one of these modes is available on the RX Series printers.

HALF-SPEED MODE

The half-speed mode cuts print speed to one-half of the maximum value. On FX Series printers, half-speed printing will be performed at 80 cps. On RX Series printers, half-speed printing will be performed at 50 cps. Half-speed printing will reduce the amount of noise that the printer generates during operation.

The ESC s code controls the half-speed mode. The BASIC expression for ESC s to enable half-speed printing is:

```
CHR$(27)"s1";
```

The expression to deactivate the half-speed mode is:

```
CHR$(27)"s0";
```

Listen to the noise that your printer generates when you run the following program. The printer will beep when it changes from half-speed printing to the regular speed. The following program illustrates the use of the half-speed print mode:

```
10 LPRINT CHR$(27)"s1";
20 GOSUB 70
30 LPRINT CHR$(7);
40 LPRINT CHR$(27)"s0";
50 GOSUB 70
60 END
70 LPRINT "JUST SOMETHING TO MAKE THE "
80 LPRINT "OPERATE SO THAT YOU CAN LISTEN"
90 LPRINT "TO THE DIFFERENCE THAT SPEED "
100 LPRINT "HAS ON THE PRINT NOISE."
110 RETURN
```

The program output is:

```
JUST SOMETHING TO MAKE THE
OPERATE SO THAT YOU CAN LISTEN
TO THE DIFFERENCE THAT SPEED
HAS ON THE PRINT NOISE.
JUST SOMETHING TO MAKE THE
OPERATE SO THAT YOU CAN LISTEN
TO THE DIFFERENCE THAT SPEED
HAS ON THE PRINT NOISE.
```

Did you hear the difference? The command in line 10 turned the half-speed print mode on. The statement in line 40 deactivated the half-speed print mode.

Immediate Print Mode (FX Printers Only)

In the immediate print mode, the output is generated as soon as data is received. The print buffer is not used to construct a complete line of output. When there is no data to be output, the printer feeds the paper up so that you can read whatever has been generated thus far. The paper will automatically be fed back down when there is more data to be output.

The immediate print mode is controlled by the ESC i code. The BASIC expression to enable the immediate print mode is:

CHR$(27)"i1";

The expression to deactivate the immediate print mode is:

CHR$(27)"i0";

The following program illustrates the use of the immediate print mode:

```
10 LPRINT CHR$(27)"i1";
20 STOP
30 LPRINT "A MESSAGE";
40 STOP
50 LPRINT " ANOTHER MESSAGE";
60 STOP
70 LPRINT CHR$(27)"i0";
80 LPRINT " A LAST MESSAGE"
90 END
```

The first action caused by the program upon execution is the advancing of the paper in the printer. The screen will then display, "Break in 20". Type CONT and press the return key. The printer will pull the paper back into itself, generate some output, and advance the paper back to its original position. The following output will be visible on the paper:

```
A MESSAGE
```

The screen will display, "Break in 40". The paper movement is performed, because the printer is in the immediate print mode. The immediate print mode allows current output to be viewed without changing the current line.

Type CONT and press the Return key. The printer will again perform its paper movement and print routine. As a result, the following output will be visible on the paper:

```
A MESSAGE ANOTHER MESSAGE
```

The screen will display, "Break in 60". Type CONT and press the Return key. This time, the printer will not advance the paper after the output has been generated because the immediate print mode was deactivated. The final output should read:

```
A MESSAGE ANOTHER MESSAGE A LAST MESSAGE
```

The immediate print mode was enabled by the command in line 10. The statement in line 70 turned the immediate print mode off.

Sounding the Buzzer

Epson printers are equipped with a buzzer. The buzzer can be used to catch the user's attention or to signal the end of processing. Recall the example program for the half-speed print mode. The buzzer was used to signal the changing of the print speed.

The BEL code controls the buzzer. The buzzer will sound once for each time the BEL code is received by the printer. The BASIC expression for BEL is:

CHR$(7);

The following program illustrates the use of CHR$(7):

```
10 LPRINT CHR$(7);
20 END
```

There is no printed output from this program. The printer should:

```
<<BEEP>>
```

Controlling The High Order Bit (FX Printers Only)

The modes covered in this section provide help for users of computers that control only seven data lines on their parallel printer interface. Examples of this type of computer are the Apple II+ and the TRS-80. This section should be skipped by owners of computers that control all eight of the data lines on their parallel printer interface.

Epson printers were designed to function with eight active data lines on the host computer's parallel printer interface. Each data line can be called a bit. If the computer sends only 7 bits, the data line that is not controlled is the one called the high order bit. Each bit has a value. The value of the high order bit is 128. Generally, the printer's high order bit is turned off. Therefore, without the high order bit, a computer can only send ASCII codes from 0 to 127. Recall that ASCII codes usually range between 0 and 255. The FX Series allow the printer's high order bit to be permanently turned on or off. Note that when the high order bit is turned on or off, the printer ignores any data sent via the eighth data line.

Activating the high order bit allows ASCII codes 128 through 255 to be sent to the printer. Note that when the high order bit is turned on, the printer adds 128 to the value of each ASCII code as it is received. Therefore, the high order bit should be activated only when an ASCII code value greater than 127 needs to be sent. The high order bit should be returned to off immediately after the ASCII code value greater than 127 has been sent.

The control codes on the FX Series are extended so that the printer will recognize a control code even when 128 is added by the high order bit. This extension allows the selection of printer modes even when the high order bit is turned on.

For example, when the high order bit is turned on, the printer interprets,

CHR$(27)"M";

as the message to turn on the elite mode just as it would if the high order bit was not turned on.

The ESC > code activates the high order bit. The ESC = code deactivates the high order bit. When either of these modes are active, the printer ignores the eighth data line. The high order bit is set either on or off. The ESC # code returns the control of the high order bit to the interface. That is, the printer uses the value sent via the eighth data line to determine the setting of the high order bit. The ESC # code is equivalent to the ESC = code for computers that control only seven of the data lines on the parallel printer interface.

The BASIC expression for ESC > is:

CHR$(27)">";

The expression for ESC = is:

CHR$(27)"=";

And, the expression for ESC # is:

CHR$(27)"#";

The following example program illustrates the use of the codes controlling the high order bit:

```
10 GOSUB 120
20 LPRINT CHR$(27)"M";
30 GOSUB 120
40 LPRINT CHR$(27)">";
50 GOSUB 120
60 LPRINT CHR$(27)"P";
70 GOSUB 120
80 LPRINT CHR$(27)"=";
90 GOSUB 120
100 LPRINT CHR$(27)"#";
110 END
120 FOR I=193 TO 196
130 LPRINT CHR$(I);
140 NEXT I
150 LPRINT
160 RETURN
```

The program output is:

```
ABCD
ABCD
ABCD
ABCD
ABCD
```

The output was generated by the subroutine beginning at line 120. This subroutine sends the ASCII codes 193 through 196 to the printer. These codes correspond to the italicized characters A through D. Notice that the first two lines of output were generated using the Roman character set. Since a computer that controls only seven data lines on the parallel printer interface cannot send a 128 corresponding to the high order bit, the codes sent by the subroutine ranged from 65 to 68. The high order bit was lost, which is equivalent to subtracting 128 from the ASCII code of each character. ASCII codes 65 through 68 correspond to Roman characters A through D.

The LPRINT statement in line 40 turned the high order bit on. This resulted in the third and fourth lines of output being generated using the italic character set. ASCII codes 65 through 68 were still received by the printer. But, since the high order bit was on, 128 was added to each code as it was received.

Take special note of the statement in line 60. We were able to use the same control code as usual, even though the high order bit was turned on. The printer still recognized the control code with the 128 added to each character. This recognition is the advantage of the extended control codes.

The command in line 80 turned the high order bit off. Notice that the fifth line of output was generated using Roman characters. Once again, the ASCII codes 65 through 68 were output instead of 193 through 196. The statement in line 100 returned the high order bit to normal.

9

Line Spacing and Line Feeds

Introduction

In this chapter, we will discuss the concepts of line spacing and line feeding. A **line** is a set of spaces across the page. Each space is nine dots in height and one character in width. The length of the line is determined by the number of spaces in the line and the pitch mode. Table 9.1 shows the maximum number of spaces in a line for each of the six pitch modes. Output can be generated only in a horizontal fashion on one line. The print head can be moved to any space along the line. A representation of a line is shown in figure 9.1.

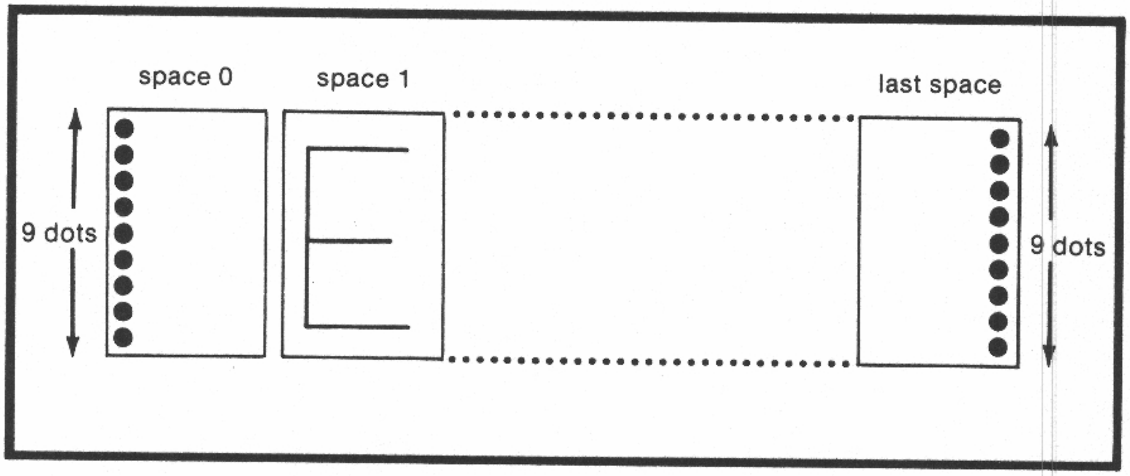


FIGURE 9.1. Representation of a line

A line feed changes the current line by moving the paper. The amount of paper fed through the printer is determined by the line spacing. Line spacing is critical because it affects the readability of text and the visual impact of graphics.

We will also discuss the carriage return in this chapter. A carriage return sets the next print position to the first space on the line. The carriage return is included with the discussion of line spacing, because the execution of a carriage return and line feed is used to access a new line.

Table 9.1. Maximum number of spaces across the page for the different pitch modes

MODEL	PITCH MODE					
	NORMAL			EXPANDED		
	Pica	Elite	Compressed	Pica	Elite	Compressed
80	80	96	137	40	48	68
100	136	163	233	68	81	116

Standard Line Spacing

1/6 INCH

The default value for line spacing is 1/6 of an inch. This line spacing provides six lines per inch.

The printer uses nine rows of dots to form characters. A line spacing of 1/6 inch is equivalent to 12 rows of dots. So, 1/6 of an inch provides a blank space equivalent to three dots between each line. This spacing is most often used to output text.

The ESC 2 code restores 1/6 inch line spacing when another line spacing is active. The BASIC expression for ESC 2 is:

CHR$(27)"2";

The following program illustrates 1/6 inch line spacing:

```
10 FOR I=1 TO 3
20 LPRINT "THIS IS 1/6 INCH LINE SPACING."
30 NEXT I
40 END
```

The program output is:

```
THIS IS 1/6 INCH LINE SPACING.
THIS IS 1/6 INCH LINE SPACING.
THIS IS 1/6 INCH LINE SPACING.
```

Notice that no command was necessary to enable 1/6 inch spacing. No other line spacing was active, so the output was generated with 1/6 inch line spacing.

1/8 INCH

Another standard line spacing is 1/8 of an inch. This line spacing provides eight lines per inch.

A line spacing of 1/8 inch is equivalent to 9 rows of dots. Therefore, 1/8 inch line spacing provides no blank space between lines. This spacing is most often used when generating graphics.

The ESC 0 code activates 1/8 inch line spacing. The BASIC expression for ESC 0 is:

CHR$(27)"0";

The following program illustrates the use of 1/8 inch line spacing:

```
10 LPRINT CHR$(27)"0";
20 FOR I=1 TO 2
30 LPRINT "THIS IS 1/8 INCH LINE SPACING."
40 LPRINT "gggjjjqqqpppgggjjjqqqppp"
50 NEXT I
60 LPRINT CHR$(27)"2"
70 FOR I=1 TO 2
80 LPRINT "THIS IS 1/6 INCH LINE SPACING."
90 LPRINT "gggjjjqqqpppgggjjjqqqppp"
100 NEXT I
110 END
```

The program output is:

```
THIS IS 1/8 INCH LINE SPACING.
gggjjjqqqpppgggjjjqqqppp
THIS IS 1/8 INCH LINE SPACING.
gggjjjqqqpppgggjjjqqqppp

THIS IS 1/6 INCH LINE SPACING.
gggjjjqqqpppgggjjjqqqppp
THIS IS 1/6 INCH LINE SPACING.
gggjjjqqqpppgggjjjqqqppp
```

The statement in line 10 activated the 1/8 inch line spacing. The command in line 60 returned the printer to 1/6 inch line spacing.

Notice the difference between 1/6 and 1/8 inch line spacing. With 1/8 inch spacing, the lower case letters with descenders below the normal print line (i.e. g, j, q, p) almost touch the top of the next line of output. The 1/6 inch spacing provides for an area with no printing between each line even when lower case letters with descenders below the normal print line are output.

7/72 INCH

The last standard line spacing is 7/72 of an inch. A line spacing of 7/72 of an inch is equivalent to 7 rows of dots. With 7/72 inch line spacing, some characters will overlap on subsequent lines. So, 7/72 inch is generally not used to generate text. This spacing is quite useful when generating graphics.

The ESC 1 code enables 7/72 inch line spacing. The BASIC expression for ESC 1 is:

CHR$(27)"1";

The following program illustrates the use of the ESC 1 code:

```
10 LPRINT CHR$(27)"1";
20 LPRINT "gjpq gjpq gjpq"
30 LPRINT "7/72 INCH SPACING"
40 LPRINT CHR$(27)"2"
50 LPRINT "1/6 AGAIN"
60 END
```

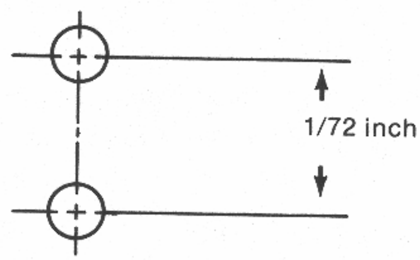
The program output is:

```
gjpq gjpq gjpq
7/72 INCH SPACING

1/6 AGAIN
```

The command in line 10 activated 7/72 inch line spacing. The statement in line 40 returned the line spacing to 1/6 inch. Notice how the descenders from the first line of output overlap the second line of output.

Variable Line Spacing

Line spacings other than the three standard spacings are available. These spacings are most useful when executing graphics. The variable line spacing is available in increments of 1/72 of an inch. The increment is significant, because the space between the centers of any two adjacent dots on the print head is 1/72 of an inch:



The ESC A code controls the variable line spacing. The BASIC expression for ESC A has the following form:

CHR$(27)"A"CHR$(*b*);

where *b* is a number or numerical expression ranging from 0 to 255. The value of *b* specifies a line spacing of *b*/72 inches. Since 1/72 inch is the height of a single dot, *b* represents the number of dots each line feed skips. The following program illustrates the use of ESC A code:

```
10 FOR I=1 TO 3
20 LPRINT "THIS IS 1/6 INCH LINE SPACING."
30 NEXT I
40 LPRINT CHR$(27)"A"CHR$(24);
50 FOR I=1 TO 3
60 LPRINT "THIS IS 1/6 INCH DOUBLE-SPACED."
70 NEXT I
80 LPRINT CHR$(27)"2";
90 FOR I=1 TO 2
100 LPRINT "1/6 AGAIN."
110 NEXT I
120 END
```

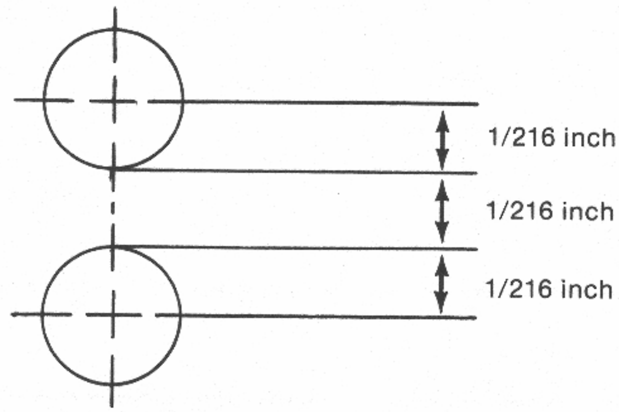
The program output is:

```
THIS IS 1/6 INCH LINE SPACING.
THIS IS 1/6 INCH LINE SPACING.
THIS IS 1/6 INCH LINE SPACING.
THIS IS 1/6 INCH DOUBLE-SPACED.

THIS IS 1/6 INCH DOUBLE-SPACED.

THIS IS 1/6 INCH DOUBLE-SPACED.

1/6 AGAIN.
1/6 AGAIN.
```

Notice that changing the line spacing allows for double-space output without having to insert a blank LPRINT command or issue extra line feeds. The statement in line 40 changed the line spacing. The value of *b* was 24, which corresponds to a line spacing of 2/6 inch. The command in line 80 returned the line spacing to the original 1/6 inch.

Microscopic Line Spacing

Microscopic line spacing is similar to variable line spacing. This spacing allows the user to specify the spacing in increments of 1/216 of an inch. This increment is significant because it is the distance between the bottom of one and the top of the next subsequent dot.



Microscopic line spacing is most useful when doing graphics.

The ESC 3 code controls microscopic line spacing. The BASIC expression for ESC 3 has the following form:

CHR$(27)"3"CHR$(*b*);

where *b* is a number or numerical expression ranging from 0 to 255. The value of *b* specifies a line spacing of *b*/216 inches. The following program illustrates the use of ESC 3 code:

```
10 FOR I=3 TO 5
20 LPRINT CHR$(27)"3"CHR$(1);
30 FOR J=1 TO I
40 LPRINT I"-STRIKE PRINTING."
50 NEXT J
60 LPRINT CHR$(27)"2"
70 NEXT I
80 LPRINT "NORMAL AGAIN"
90 END
```

The program output is:

```
3 -STRIKE PRINTING.
4 -STRIKE PRINTING.
5 -STRIKE PRINTING.
NORMAL AGAIN
```

This program manually imitates the double-strike mode. The command in line 20 sets the line spacing to 1/216 of an inch. A multiple strike line is then generated by the "J" FOR...NEXT loop in lines 30 through 50. The statement in line 60 changes the line spacing back to the standard 1/6 inch. Because the LPRINT command used in line 60 does not end in a semicolon, a carriage return and a line feed pair were issued. this pair caused each of the multiple strike lines to be output on a new line.

Carriage Return

As mentioned in the introduction, a carriage return sets the next print position to the first space of the line. The LPRINT command will automatically handle the sending of the carriage return to the printer. However, carriage returns can also be sent manually.

The CR code controls the carriage return. The BASIC expression for CR is:

CHR$(13);

The following program illustrates the use of the carriage return:

```
10 LPRINT "SOMETHING IN THE BUFFER.";
20 LPRINT CHR$(13);
30 LPRINT "SOMETHING ELSE IN THE BUFFER."
40 END
```

The output is:

```
SOMETHING ENSEHENBUREEBUFFER.
```

This output looks rather strange. The command in line 10 sent the following to the print buffer:

```
SOMETHING IN THE BUFFER.
```

The statement in line 20 issued a carriage return. However, no line feed was issued. So, the following,

```
SOMETHING ELSE IN THE BUFFER.
```

was output when the command in line 30 was executed. Since no paper movement had occurred, this second line was printed directly on top of the first.

One-Time Line Feed

A line feed causes the paper to be advanced by the amount specified by the line spacing presently in force. The LPRINT command will automatically handle sending of the line feed to the printer. However, line feeds can also be sent manually.

The LF code controls the line feed. The BASIC expression for LF is:

CHR$(10);

The following program illustrates the use of LF code:

```
10 LPRINT "SOMETHING IN THE BUFFER.";
20 LPRINT CHR$(13);
30 LPRINT CHR$(10);
40 LPRINT "SOMETHING ELSE IN THE BUFFER."
50 END
```

The program output is:

```
SOMETHING IN THE BUFFER.
SOMETHING ELSE IN THE BUFFER.
```

Notice that this example program is identical to the preceding one except for the addition of line 30. When the line feed is sent to the printer by the command in line 30, the program outputs the data correctly.

One-Time Variable Line Feed

The variable line feed causes paper to be advanced by an amount specified in the code. The amount to be advanced can be specified in increments of 1/216 of an inch. This line feed will be performed only once and will use the internally defined variable spacing. The variable line feed does not change the line spacing presently in force.

The ESC J code controls the variable line feed. The BASIC expression for ESC J has the form:

CHR$(27)"J"CHR$(*b*);

where *b* represents a number or numerical expression ranging from 0 to 255. The value of *b* specifies a line spacing of *b*/216 inches. The following program illustrates the use of the variable line feed:

```
10 LPRINT "SOMETHING";
20 LPRINT CHR$(27)"J"CHR$(216);
30 LPRINT " AND THE NEXT";
40 LPRINT CHR$(27)"J"CHR$(4);
50 LPRINT " AND THE LAST"
60 END
```

The program output is:

```
SOMETHING





                AND THE NEXT AND THE LAST
```

Notice that no carriage return is issued with the variable line feed. The command in line 20 caused a one inch space between the first and second lines of output. The statement in line 40 caused a small (4/216 inch) line feed between the second and the last lines of output.

Reverse Line Feed

The reverse line feed is available on the FX Series printers only. The reverse line feed acts exactly like the variable line feed, except that the paper is moved backward instead of being advanced.

Never use the reverse line feed when the printer is loaded with gummed labels. If you do, the labels will tend to peel off of their backing and stick inside of the printer. Such a disaster could seriously jam the printer.

The ESC j code controls the reverse line feed. The BASIC expression for ESC j has the form:

CHR$(27)"j"CHR$(*b*);

where b represents a number or numerical expression ranging from 0 to 255. The value of b specifies a line spacing of $b/216$ inches. The following program illustrates the use of the reverse line feed:

```
10 LPRINT "START ";
20 LPRINT CHR$(27)"j"CHR$(50);
30 LPRINT "AND UP";
40 LPRINT CHR$(27)"j"CHR$(25);
50 LPRINT " AND UP";
60 LPRINT CHR$(27)"J"CHR$(75);
70 LPRINT " AND BACK DOWN"
80 END
```

The program output is:

```
                    AND UP
          AND UP

START                          AND BACK DOWN
```

Note the order in which the output was generated. START was printed first. Then, the first AND UP was printed, followed by the next AND UP. AND BACK DOWN was printed last. The commands in both lines 20 and 40 caused reverse paper feeds of different lengths. The command in line 70 caused the paper to be repositioned to the original starting line.

10

Controlling Forms

Introduction

In this chapter, we will discuss the various control codes that can be used to regulate the movement of forms through the Epson printer. A number of different types of forms can be used with the Epson including:

- Fanfold paper
- Invoices
- Gummed labels
- Checks

These are depicted in figure 10.1.

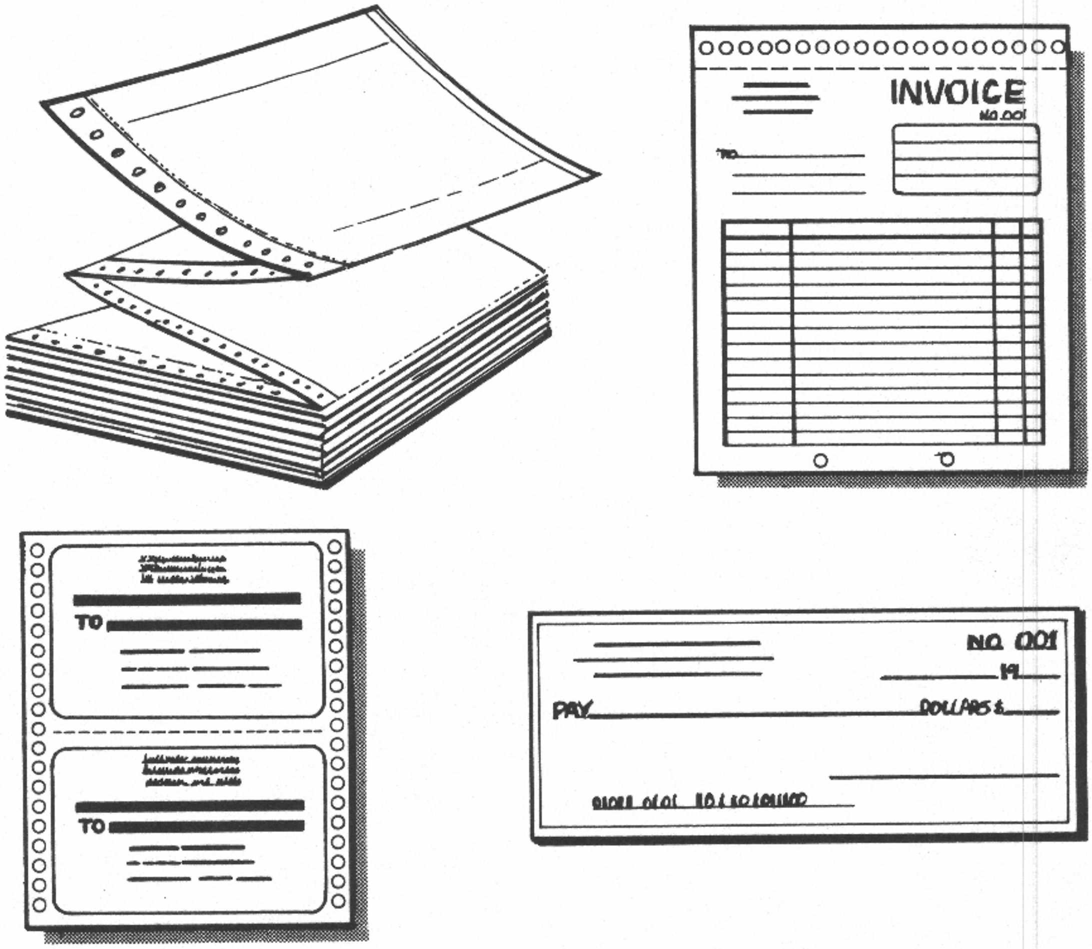


FIGURE 10.1. Forms that can be used with the Epson

Top-of-Form

The top-of-form position is the site of the first print line on a form. The top-of-form position does not have to coincide with the first line below the perforation between two forms. Defining the top-of-form position a few lines into the form allows a top margin to be created each time a new form is accessed. However, you should always be certain that you set the top-of-form position before using any of the control codes relating to forms.

To set the top-of-form position, first place the site of the first print line directly beneath the print head. The simplest method of accomplishing this placement is to use the manual feed knob to align the first print line under the print head. After the paper has been aligned, that position

can be set as the top-of-form by either switching the printer off and then back on, or by sending the Master Reset Code to the printer. Note that both of these methods reset all of the printer modes to their default values. Therefore, be sure to set the top-of-form position before enabling any modes.

Sending a Form Feed

A form feed causes the paper to be advanced so that the next top-of-form position lies directly beneath the print head. Before using the form feed, verify that the top-of-form position has been set correctly.

The FF code controls the form feed. The BASIC expression for FF is:

```
CHR$(12);
```

The use of the form feed is illustrated in the example programs in the following sections.

Changing the Form Length

The form length can be defined as the amount of space between the top of one form and the top of the subsequent form. The default form length is eleven inches. If a form with a different length is being used, the form length must be changed to accommodate the new form.

The form length can be specified either as a number of inches or as a number of lines. If the form length is specified as a number of lines, the length will be converted to a number of inches. This form length is calculated by the following formula:

length in inches = (length in lines) x (current line spacing)

When a length is defined in this fashion, it is called an absolute length. The absolute length will not be changed even if a different line spacing is selected.

The ESC C code is used to control the form length setting. The BASIC expression to define form length as a number of inches has the following form:

CHR$(27)"C"CHR$(0)CHR$(*b*);

where *b* represents a number or numerical expression from 1 to 22. The value of *b* specifies the length of the form in inches.

The BASIC expression to define form length as a number of lines has the following form:

CHR$(27)"C"CHR$(*b*);

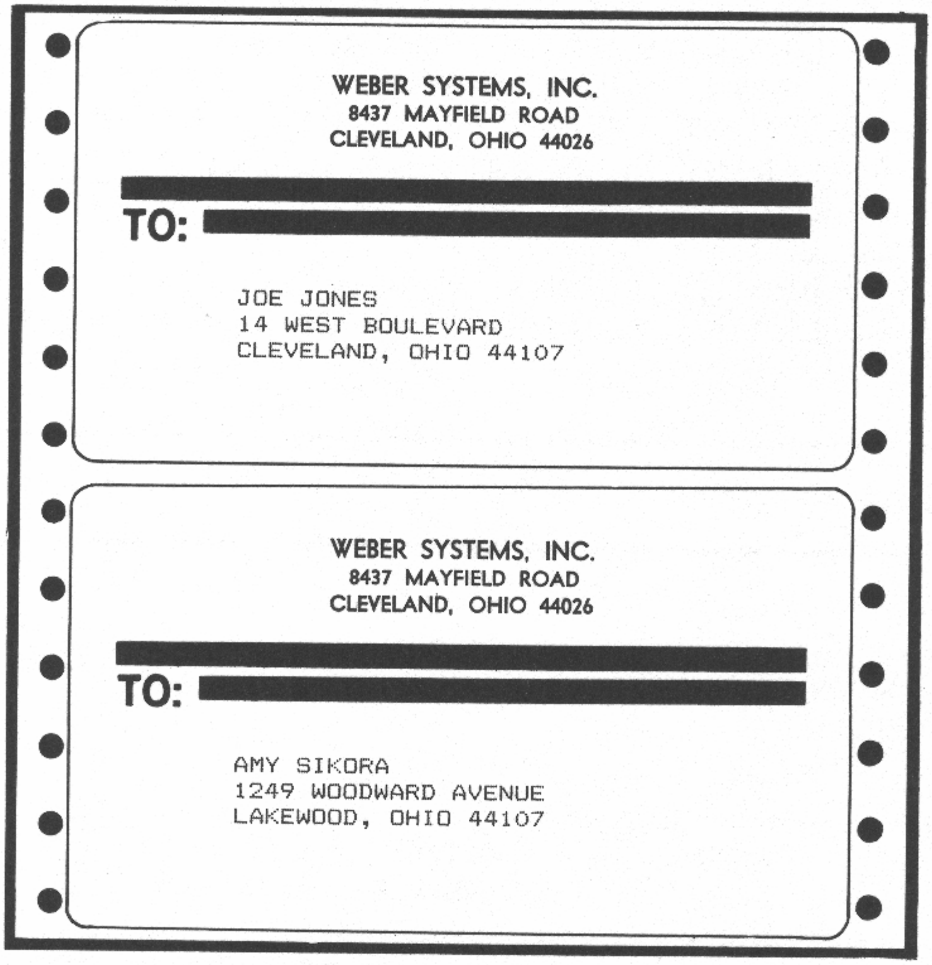
where *b* represents a number or numerical expression from 1 to 127. The value of *b* specifies the length of the form in lines. The printer will convert the number of lines to an absolute length in inches.

The first example program illustrates how to specify form length as a number of inches. In this program, mailing labels were used as the form. The second example program shows how to specify form length as a number of lines. Regular 8½" x 11" paper is used in the second example.

The first example program is:

```
10 LPRINT CHR$(27)"C"CHR$(0)CHR$(3);
20 LPRINT "JOE JONES"
30 LPRINT "14 WEST BOULEVARD"
40 LPRINT "CLEVELAND, OHIO 44107"
50 LPRINT CHR$(12);
60 LPRINT "AMY SIKORA"
70 LPRINT "1249 WOODWARD AVENUE"
80 LPRINT "LAKEWOOD, OHIO 44107"
90 END
```

The program output is:



Note that the output has been reduced by 65%. The mailing labels used in this example are three inches in length. The statement in line 10 changed the form length to three inches. The command in line 50 is a form feed. Note that before the program was run, the top-of-form position was set to the site on the mailing label where the name was printed.

The second example:

```
10 LPRINT CHR$(27)"C"CHR$(5);
20 LPRINT "JOE JONES"
30 LPRINT "YEARS OF SERVICE: 10"
40 LPRINT "CURRENT PAY SCALE: 12.79/HR"
50 LPRINT CHR$(12);
60 LPRINT "AMY SIKORA"
70 LPRINT "YEARS OF SERVICE: 3"
80 LPRINT "CURRENT PAY SCALE: 8.50/HR"
90 END
```

The program output is:

```
JOE JONES
YEARS OF SERVICE: 10
CURRENT PAY SCALE: 12.79/HR


AMY SIKORA
YEARS OF SERVICE: 3
CURRENT PAY SCALE: 8.50/HR
```

The statement in line 10 changed the form length to 5 lines. The current line spacing was 1/6 inch. So the absolute form length was calculated as:

length in inches = (5 lines) x (1/6 inch/line)
length in inches = 5/6 inch

The command in line 50 is a form feed. Note that the form in this example was a set amount of space, not an actual page.

Perforation Skip-Over

Epson printer's perforation skip-over feature allows the user to avoid generating output on or near the perforation between two forms. The amount of space to be skipped is specified as a number of lines. As discussed in the section on form length, the number of lines is converted to an absolute length. Recall that an absolute length is not affected by a change in the line spacing.

Before the skip-over feature can be used, the top-of-form position must be set to conform with the first line of the present form. Also, the form length must be set to the length of the form being used. If these two values are not set correctly, the skip will occur at a position other than the perforation.

The ESC N code enables the perforation skip-over feature. The ESC O code deactivates the skip-over feature. Changing the form length with ESC C will also turn the perforation skip-over feature off. The BASIC expression for ESC N has the form:

CHR$(27)"N"CHR$(*b*);

where *b* represents a number or numerical expression from 1 to 127. The value of *b* specifies the number of lines to be skipped at the end of the form. If the value of *b* is greater than the number of lines on the form presently in use, the skip-over function will be ignored. The BASIC expression for ESC O is:

CHR$(27)"O";

The following program illustrates the use of the perforation skip-over feature:

```
10 LPRINT CHR$(27)"N"CHR$(6);
20 LPRINT "THIS IS THE LAST LINE OF THE FORM"
30 LPRINT "THIS IS THE TOP OF THE NEXT FORM"
40 LPRINT CHR$(27)"O";
```

The program output is:

```
THIS IS THE LAST LINE OF THE FORM

-----------------------------------------

THIS IS THE TOP OF THE NEXT FORM
```

Before running this program, we performed several steps. First, we set the top-of-form position three lines below the perforation. We then checked to be sure that the defined form length coincided with the actual length of one form. Finally, we positioned the paper so that the current line was seven lines before the end of the form.

The statement in line 10 defined the length of the perforation skip-over to be 6 lines and enabled the skip-over feature. Since the line spacing was 1/6 inch, the absolute length of the skip was calculated as one inch. The LPRINT command in line 20 generated the first line of output. The line feed added to the data by the LPRINT command caused the current line to be changed to 6 lines before the end of the form. Since the current line was now in the area to be skipped, the paper was fed to the top of the next form.

The statement in line 30 generated the output at the top of the next form. Notice that defining the top-of-form position a few lines into the form allows the perforation skip-over to construct both a bottom and a top margin at the same time.

Controlling the Paper-Out Sensor

When the paper-out sensor is active and the end of the paper is detected, the printer falls off line, sounds the buzzer, and lights the paper-out indicator. The paper-out sensor prevents the print head from striking the roller. But, the sensor also prevents use of the last quarter of the page. This is generally only a minor annoyance when using a continuous type of paper such as fanfold. However, with single-sheet paper, not

being able to use the last quarter of the page could be a major problem. This problem can be averted by deactivating the paper-out sensor.

If single-sheet paper is always used, the paper-out sensor can be permanently deactivated by setting the DIP switches as discussed in chapter 13. The sensor can be deactivated for only a short period of time by using the ESC 8 code to turn the paper-out sensor off. The ESC 9 code enables the sensor. The BASIC expression for ESC 8 is:

```
CHR$(27)"8";
```

The expression for ESC 9 is:

```
CHR$(27)"9";
```

For a program example using ESC 8 and ESC 9, see the single-sheet example in chapter 12.

11

Formatting Output on the Page

Introduction

Output can be formatted horizontally and vertically on each form. Horizontal formatting involves controlling the placement of the output across the page. Constructing columns of data requires horizontal formatting. Vertical formatting controls the placement of output down the page. Creating several rows of data at different sites on the page involves vertical formatting.

To enable formatting of the output, the printer's tabs are used. Generally, these tabs will be easier to use than BASIC's TAB command. One reason for this is that the TAB command was designed primarily for use with the display screen rather than with a printer. Also, the manner in which the TAB command functions may vary with the version of BASIC in use. One final advantage is that the printer's tabs can automatically accommodate for the different print pitch modes.

Vertical Tabs

Vertical tabs are used to control the vertical formatting of output on a page. The older RX Series use an entirely different scheme to accomplish vertical tabulation than the newer RX Series and the FX Series. If you have an RX Series printer, consult the manual to see which tab scheme your printer utilizes. The older models use the ESC e code to control setting of the tab stops. We have dedicated a separate section describing vertical tabs for each scheme.

VERTICAL TABS — OLDER RX SERIES

To accomplish vertical tabulation, the older RX series sets tab stops at regular intervals down the page. For example, if the interval is 5, the first tab stop is set at the fifth line. Subsequent stops are set every 5 lines (i.e. 10, 15, 20,...). The last tab stop occurs at the interval closest to the end of the form without exceeding the form length. For the previous example, if the form length were 66 lines, the last tab stop would be set at 65. Remember that the line numbering begins with line 0 and, in this case, ends with line 65.

Tab stops are stored as absolute lengths. That is, the current line spacing is used to find the distance of each stop from the top of the form in inches. For example, suppose the 1/6 inch line spacing were currently in force and a tab interval of 5 were specified. The position of the first tab stop would be calculated as:

Position = (1/6 inch/line) x (5 lines)
Position = 5/6 inch

The position of all the other stops would be calculated in a similar fashion. So, the position of the tab stop at line 20 would be:

Position = (1/6 inch/line) x (20 lines)
Position = 3-1/3 inches

These absolute lengths are then used to execute vertical tabs. Changing the line spacing will not change the position of the tab stops on the page.

Once the tab stops have been set, each time a vertical tab is executed, the printer will advance the paper so that the line corresponding to the next tab is positioned beneath the print head. In the previous example, if the current line on the form was line 8, a vertical tab would cause the current line to be changed to line 10. A second vertical tab would cause the current line to be changed to line 15.

The ESC e1 code controls setting of the vertical tab stops. Input of the VT code causes a vertical tab to be executed. The BASIC expression for ESC e1 has the form:

CHR$(27)"e1"CHR$(*b*);

where *b* represents a number or numerical expression ranging from 0 to the form length. The value of *b* specifies the vertical tab stop interval. If the value of *b* is zero, execution of the VT code causes no paper movement. If the value of *b* is greater than the form length, the setting is ignored. The BASIC expression for VT is:

CHR$(11);

If no tab stops are defined, the VT code acts exactly like a line feed. The following program illustrates the use of vertical tabulation:

```
10 LPRINT CHR$(27)"e1"CHR$(3);
20 LPRINT "LINE ZERO"
30 LPRINT CHR$(11);
40 LPRINT "LINE THREE"
50 LPRINT CHR$(11)CHR$(11)CHR$(11);
60 LPRINT "LINE TWELVE"
70 END
```

The program output is:

```
LINE ZERO

LINE THREE







LINE TWELVE
```

Before the program was run, we forced the print head to the top-of-form position by turning the printer off and then back on. The command in line 10 set the tab stops at every third line beginning with line 3. The command in line 30 caused a vertical tab to line 3 on the form. Notice that two lines separate the form's first line (line 0) and line 3. The statement in line 50 caused three vertical tabs. These tabs caused the current line to be changed to line 12.

VERTICAL TABS — NEWER RX AND FX SERIES

To accomplish vertical tabulation, tab stops must first be defined on the form. As many as 16 vertical tabs stops can be defined. Once the vertical tabs have been defined, each vertical tabulation causes the paper to be advanced until the position of the next tab stop lies directly beneath the print head.

Tab stops are stored as absolute lengths. That is, the current line spacing is used to find the distance of each stop from the top of the form in inches. For example, suppose the 1/6 inch line spacing were currently in force and tab stops were defined at lines 5 and 20. The position of the first tab stop would be calculated as:

Position = (1/6 inch/line) x (5 lines)
Position = 5/6 inch

The position of the other stop would be calculated in a similar fashion. So, the position of the tab stop at line 20 would be:

Position = (1/6 inch/line) x (20 lines)
Position = 3-1/3 inches

These absolute lengths are then used to execute vertical tabs. Changing the line spacing will not change the position of the tab stops on the page.

The ESC B code is used to define the tab stops. Input of the VT code will cause a vertical tab to be executed. The BASIC expression for ESC B has the form:

CHR\$(27)"B" CHR\$(S_1) CHR\$(S_2) ... CHR\$(S_{16}) CHR\$(1);

where S_1, S_2,...,S_{16} are numbers or numerical expressions. Each S value defines a tab stop. The value of S represents the line on the form where a tab stop should be set. These S values must be arranged in order from lowest to highest. The CHR\$(1) is the end marker. When all the desired vertical tab stops have been defined, add the end marker to the tail of the list. Note that all sixteen of the S values need not be defined. The BASIC expression for VT is:

CHR\$(11);

The following program illustrates the use of vertical tabs:

```
10 LPRINT CHR$(27)"B"CHR$(3)CHR$(12)CHR$(1);
20 LPRINT "LINE ZERO"
30 LPRINT CHR$(11);
40 LPRINT "LINE 3"
50 LPRINT CHR$(11);
60 LPRINT "LINE 12"
70 END
```

The program output is:

```
LINE ZERO

LINE 3




LINE 12
```

Before the program was run, the paper was positioned at the top-of-form position. Notice that the first line of the form is numbered as line 0. The statement in line 10 caused tab stops to be set at lines 3 and 12 on the form. The commands in lines 30 and 50 caused the vertical tabs.

Saving Vertical Tab Stops in Channels (Newer RX and FX Printers Only)

Newer RX Series and FX Series printers have an additional feature that allows a set of vertical tab stops to be saved for later use. One set of tab stops is called a **channel**. This feature is valuable when the printer is being used to output multipage reports or documents where each page has a different vertical format.

The ESC b code controls the setting of a channel. A different channel can be selected using the ESC / code. The BASIC expression for ESC b has the form:

CHR\$(27) "b" CHR\$(n) CHR\$(S_1) CHR\$(S_2) ... CHR\$(S_{16}) CHR\$(1);

where n and S_1, S_2, ... , S_{16} are numbers or numerical expressions. The n value specifies the reference number for this channel of tab stops. The value of n can range from 0 to 7. The S values are exactly identical to the S values in the ESC B code discussed in the previous section. In fact, an ESC b code with a reference number of zero is the same as the ESC B code. The BASIC expression for ESC / has the form:

CHR$(27) "/" CHR$(*n*);

where n is a number or numerical expression ranging from 0 to 7. The value of n specifies which channel is to be selected. For an example program illustrating the use of vertical tab stops saved in channels, see the single-sheet example program in chapter 12.

Horizontal Tabs

Horizontal tabs are used to control the vertical formatting of output on a page. The older RX Series use an entirely different scheme to accomplish horizontal tabulation than the newer RX Series and the FX Series. If you have an RX Series printer, consult the manual to see which tab scheme your printer utilizes. The older models use the ESC e code to control setting of tab stops. We have dedicated a separate section describing horizontal tabs for each scheme.

HORIZONTAL TABS — OLDER RX SERIES

To accomplish horizontal tabulation, the RX Series sets tab stops at regular intervals across the page. For example, if the interval is 5, the first tab stop will be set at the fifth column. Subsequent stops will be set at every 5 columns (i.e. 10, 15, 20,...). The last tab stop is the interval closest to the right margin without exceeding the line length. For example, if the line length was 80 columns, the last tab stop would be set at 75. Remember that the column numbering begins with column zero and, in this case, ends with column 79.

These tab stops are stored as absolute lengths. That is, they are stored as a number of inches instead of a number of columns. So, changing the print pitch will not change the placement of the tab stop on the page.

Once the tab stops have been set, each time a horizontal tab is executed, the printer will change the next print position to the column corresponding to the next tab stop to the right on the page. In the previous example, if the next print position were column 22, a horizontal tab would cause the next print position to change to column 25.

The ESC e0 code controls the setting of the horizontal tab stops. Input of the HT code causes a horizontal tab to be executed. The BASIC expression for ESC e0 has the form:

CHR$(27) "e0" CHR$(*b*);

where *b* represents a number or numerical expression from 0 to the maximum allowable value. These values are listed in table 11.1. Note that the maximum value is different for each pitch. The value of *b* specifies the horizontal tab stop interval. If the value of *b* is zero, a horizontal tab will have no effect. If you do not define tab stops, the printer will use a default interval of 8 to execute horizontal tabs. The BASIC expression for the HT code is:

CHR$(9);

Table 11.1. Maximum tab interval for the six print pitches

Model	PRINT PITCH					
	Pica	Enlarged Pica	Elite	Enlarged Elite	Compressed	Enlarged Compressed
80	21	10	25	12	36	18
100	44	22	54	27	77	38

The following program illustrates the use of horizontal tabs:

```
10 LPRINT "012345678901234567890123456789O"
20 LPRINT CHR$(27)"e0"CHR$(5);
30 LPRINT CHR$(9);
40 LPRINT "NAME"CHR$(9)CHR$(9);
50 LPRINT "ITEM 1"CHR$(9)"ITEM 2"
60 END
```

The program output is:

```
     012345678901234567890123456789O
          NAME          ITEM 1      ITEM 2
```

The statement in line 20 set the horizontal tab interval at 5 columns. The command in line 30 caused a horizontal tab to column 5. The statement in line 50 caused two horizontal tabs that relocated the next print position to column 15. Finally, the command in line 70 caused only one horizontal tab that changed the next print position to column 25. Note that the printing of the data "ITEM 1" moved the next print position to column 21 before the horizontal tab to column 25 was executed. The tab stop at column 20 was never used because the next print position had been moved past the tab stop when "ITEM 1" was printed.

HORIZONTAL TABS — NEWER RX AND FX SERIES

To accomplish horizontal tabulation, tab stops must first be defined across the form. As many as 32 horizontal tab stops can be defined across the form. Each time a horizontal tab is executed, the printer will change the next print position to the column corresponding to the next tab stop to the right on the page.

Note that horizontal tab stops are stored as absolute lengths. Therefore, changing the print pitch will not change the position of the tab stops across the page.

The ESC D code is used to define the tab stops. Input of the HT code will cause a horizontal tab to be executed. The BASIC expression for ESC D has the form:

CHR$(27)"D" CHR$(S_1) CHR$($S_2$)...CHR$(S_{32}) CHR$(1);

where S_1, S_2,..., S_{32} are numbers or numerical expressions ranging from 0 to the last column on the page. Table 11.2 lists these maximum *S* values. The value of *S* represents the column across the form where a tab stop should be set. These *S* values must be arranged in order from lowest to highest. CHR$(1) is the end marker. When all desired horizontal tab stops have been defined, the end marker should be added to the tail of the list. Note that all 32 *S* values need not be defined. The BASIC expression for HT is:

CHR$(9);

Table 11.2. Maximum horizontal tab stop setting for the six print pitches

	PRINT PITCH					
Model	**Pica**	**Enlarge Pica**	**Elite**	**Enlarged Elite**	**Compressed**	**Enlarged Compressed**
80	79	39	95	47	131	65
100	135	67	163	80	232	115

The following program illustrates the use of horizontal tabs:

```
10 LPRINT "0123456789012345678901234567890"
20 LPRINT CHR$(27)"D"CHR$(5)CHR$(15)CHR$(25);
30 LPRINT CHR$(1);
40 LPRINT CHR$(9);
50 LPRINT "NAME";
60 LPRINT CHR$(9);
70 LPRINT "ITEM 1";
80 LPRINT CHR$(9);
90 LPRINT "ITEM 2";
100 END
```

The program output is:

```
  01234567890123456789012345678901
       NAME      ITEM 1    ITEM 2
```

The statement in lines 20 and 30 set the horizontal tab stops. The stops were set to columns 5, 15, and 25. The CHR$(1) in line 30 was the end marker. The command in line 40 caused a horizontal tab to column 5. The statement in line 60 caused a tab to column 15. The command in line 80 caused a horizontal tab to column 25.

Setting Margins

The margin settings determine the number of columns across the page. The left margin setting defines the horizontal position on the page where output will begin for each line. The right margin marks the horizontal position on the page where output will end for each line. Both margins settings have maximum and minimum allowable values in each of the six pitch modes. These values are listed in table 11.3. This table also details the minimum number of columns across the page that can be specified. The default left margin setting is column zero for all models. For the 80 column models, the default right margin setting is column 79.

Table 11.3. Margin setting information

Model			PRINT PITCH					
			Pica	Enlarged Pica	Elite	Enlarged Elite	Compressed	Enlarged Compressed
80	Left Margin	minimum	0	0	0	0	0	0
		maximum	78	39	93	46	134	66
	Right Margin	minimum	2	1	3	2	4	2
		maximum	79	39	95	47	136	67
	Columns Across Page	minimum	2	1	3	2	4	2
		maximum	80	40	96	48	137	68
100	Left Margin	minimum	0	0	0	0	0	0
		maximum	134	67	160	79	229	114
	Right Margin	minimum	2	1	3	2	4	2
		maximum	135	68	163	81	233	116
	Columns Across Page	minimum	2	1	3	2	4	2
		maximum	136	68	163	81	233	116

Note that when the left margin is changed, all horizontal tab stops set thereafter will treat the new left margin as column 0. All previously set horizontal tab stops will be erased when either of the margins are changed. Furthermore, changing either of the margins causes the print buffer to be cleared. Be careful not to inadvertently erase some of your print data. The easiest way to avoid this mistake is to insert a blank LPRINT command before changing the margin settings.

The ESC Q code sets the right margin, and the ESC l controls the setting of the left margin. The BASIC expression for ESC Q has the form:

CHR$(27) "Q" CHR$(*b*);

where *b* represents a number or numerical expression. The range of *b* for the six pitch modes is listed in table 11.3. Use the values corresponding to the right margin. The value of *b* specifies the column at which the right margin is to be set. The BASIC expression for ESC l has the form:

CHR$(27) "l" CHR$(*b*);

where *b* represents a number or numerical expression. The range of *b* for the six pitch modes is listed in table 11.3. Use the values corresponding to the left margin. The value of *b* specifies the column at which the left margin is to be set. Note that the value:

Right Margin — Left Margin

must fall within the range listed in table 11.3 as "Columns Across Page." The following program illustrates how to set new margins:

```
10 LPRINT "BEFORE CHANGE:"
20 GOSUB 90
30 LPRINT "AFTER CHANGE:"
40 LPRINT CHR$(27)"l"CHR$(5);
50 LPRINT CHR$(27)"Q"CHR$(10);
60 GOSUB 90
70 LPRINT CHR$(27)"@";
80 END
90 LPRINT "LLLLLLLLLLLLLLL";
100 LPRINT "LLLLLLLLLLLLLLL"
110 RETURN
```

The program output is:

```
BEFORE CHANGE:
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
AFTER CHANGE:
     LLLLL
     LLLLL
     LLLLL
     LLLLL
     LLLLL
     LLLLL
```

The statement in line 40 changed the left margin to column 5. The command in line 50 caused the right margin to be changed to column 10. Notice how different the same output data looks with the shorter length lines. The statement in line 70 changed the column settings back to their default values by using the Master Reset Code.

Unidirectional Printing

The printer usually performs bidirectional printing. When bidirectional printing is performed, one line is printed from left to right. The subsequent line is printed from right to left. The third line is printed left to right, etc. When undirectional printing is performed, all lines are printed from left to right.

Undirectional printing is more precise than bidirectional printing. Printing unidirectionally will reduce the horizontal dot aberation. Unidirectional printing is slower than bidirectional printing. However, for most applications, bidirectional printing will yield quite acceptable results.

There are two methods of controlling unidirectional printing. The ESC < code causes only the next line to be printed unidirectionally. The ESC U1 code activates unidirectional printing for all subsequent lines. The ESC U0 code deactivates unidirectional printing.

The BASIC expression for ESC < is:

CHR$(27) "<";

The following program illustrates the use of ESC <:

```
10 LPRINT CHR$(27)"<";
20 LPRINT "THIS IS UNIDIRECTIONAL PRINTING."
30 LPRINT "THESE TWO LINES WERE"
40 LPRINT "     PRINTED BIDIRECTIONALLY."
50 END
```

The program output is:

```
THIS IS UNIDIRECTIONAL PRINTING.
THESE TWO LINES WERE
     PRINTED BIDIRECTIONALLY.
```

Notice that the command in line 10 enabled unidirectional printing for one line only.

The BASIC expression for the ESC U1 code to enable unidirectional printing for all subsequent lines is:

CHR$(27) "U1";

The expression for the ESC U0 code to deactivate unidirectional printing is:

CHR$(27) "U0"

The following program illustrates the use of ESC U:

```
10 LPRINT CHR$(27)"U1";
20 LPRINT "PRINTED UNIDIRECTIONALLY:"
30 GOSUB 80
40 LPRINT CHR$(27)"U0"
50 LPRINT "PRINTED BIDIRECTIONALLY:"
60 GOSUB 80
70 END
80 LPRINT "COMPARE THESE TWO IDENTICAL"
90 LPRINT "OUTPUTS FROM THE TWO DIRECTIONAL"
100 LPRINT "PRINT MODES AND SEE IF YOU CAN"
110 LPRINT "SPOT ANY DIFFERENCES. IF YOU CAN,"
120 LPRINT "YOUR EYES ARE SHARPER THAN MINE."
130 LPRINT "UNIDIRECTIONAL PRINTING IS MORE"
140 LPRINT "USEFUL IN THE BIT IMAGE MODE."
150 RETURN
```

The program output is:

```
PRINTED UNIDIRECTIONALLY:
COMPARE THESE TWO IDENTICAL
OUTPUTS FROM THE TWO DIRECTIONAL
PRINT MODES AND SEE IF YOU CAN
SPOT ANY DIFFERENCES. IF YOU CAN,
YOUR EYES ARE SHARPER THAN MINE.
UNIDIRECTIONAL PRINTING IS MORE
USEFUL IN THE BIT IMAGE MODE.

PRINTED BIDIRECTIONALLY:
COMPARE THESE TWO IDENTICAL
OUTPUTS FROM THE TWO DIRECTIONAL
PRINT MODES AND SEE IF YOU CAN
SPOT ANY DIFFERENCES. IF YOU CAN,
YOUR EYES ARE SHARPER THAN MINE.
UNIDIRECTIONAL PRINTING IS MORE
USEFUL IN THE BIT IMAGE MODE.
```

Notice that the two modes yield identical results. We recommend that unidirectional printing be reserved for applications that require a high degree of precision in horizontal placement such as characters that must line up directly beneath one another such as tables and printer graphics.

The command on line 10 enabled unidirectional printing. The statement on line 40 returned the printer to normal, bidirectional printing.

12

Examples Combining Text Modes

Introduction

Thus far, we have discussed only modes dealing with the output of text. Before we start coverage of the modes controlling bit image graphics, we will present several example programs. These example programs will show how to use the control codes already discussed to accomplish typical tasks such as generating tables and graphs and handling single-sheet paper.

Generating a Table

Printers can be used to generate tables. In this example program we will demonstrate how table 8.1 was produced. The dimensions of the table must first be determined. We wanted the table to fit on a page in this

book. So, our table could be no wider than 4⅝ inches. We allowed the length of the table to be whatever was necessary to accommodate all of the information.

The width of the column listing for all the countries was set by the longest entry, Denmark II (RX only). So, this column used 19 of the available pica character spaces across the page. The column listing the *b* values occupied another two of these spaces. Each of the 12 columns of international characters used another of the available spaces. To make the table more readable, we added one blank space between each column. These spaces occupied another 13 of the spaces. The total number of spaces came to 46 which is just under the limit.

On the vertical we wanted each row of the table to be separated by a printed horizontal line. We made some lines heavier than others to set off the heading.

The two programs are slightly different for the two different series. We included a program for both the RX and FX Series.

RX SERIES PROGRAM

The program to generate the table using an RX Series printer follows:

```
10 LPRINT CHR$(27)"E";
20 LPRINT CHR$(27)"G";
30 LPRINT CHR$(27)"-1";
40 GOSUB 360
50 LPRINT CHR$(14);
60 FOR N=1 TO 2
70 READ A$
80 LPRINT A$;
90 LPRINT CHR$(27)"4";
100 NEXT N
110 LPRINT CHR$(27)"5"
120 LPRINT CHR$(27)"F";
130 LPRINT CHR$(27)"H";
140 FOR J=0 TO 10
150 READ A$
160 LPRINT CHR$(27)"E";
170 LPRINT A$;
```

```
180 LPRINT CHR$(27)"F";
190 LPRINT CHR$(27)"R"CHR$(J);
200 LPRINT " # $ @ [ \ ] ^ ` { ¦ } ~"
210 NEXT J
220 LPRINT CHR$(27)"@"
230 END
240 DATA "COUNTRY    ","b                   "
250 DATA USA                      0
260 DATA FRANCE                   1
270 DATA GERMANY                  2
280 DATA ENGLAND                  3
290 DATA DENMARK I                4
300 DATA SWEDEN                   5
310 DATA ITALY                    6
320 DATA SPAIN                    7
330 DATA JAPAN                    8
340 DATA NORWAY(RX only)          9
350 DATA DENMARK II(RX only) 10
360 FOR I=1 TO 46
370 LPRINT "_";
380 NEXT I
390 LPRINT
400 RETURN
410 END
```

The program output is:

COUNTRY	b												
USA	0	#	$	@	[	\	]	^	`	{	¦	}	~
FRANCE	1	#	$	à	°	ç	§	^	`	é	ù	è	¨
GERMANY	2	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
ENGLAND	3	£	$	@	[	\	]	^	`	{	¦	}	~
DENMARK I	4	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
SWEDEN	5	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
ITALY	6	#	$	@	°	\	é	^	ù	à	ò	è	ì
SPAIN	7	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
JAPAN	8	#	$	@	[	¥	]	^	`	{	¦	}	~
NORWAY(RX only)	9	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
DENMARK II(RX only)	10	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü

Notice that the output just fills the width of the page. Also, all the columns line up correctly on the page. Take special note of the DATA statements in lines 240 to 350. The necessary spaces were inserted into the data to accomplish the horizontal formatting. A horizontal tab was not used because the underlined mode was enabled to generate the line between character sets. If the horizontal tab had been used, the line would have had holes in it at the spaces that were skipped over by the tab.

The statements in lines 10 to 30 selected the print mode. The subroutine called in line 40 caused the top line of the table to be output. The command in line 50 enabled the expanded print mode for one line of output. The loop in lines 60 to 100 read the table column headings and printed them. Note that the line between the column headings and the rest of the table were generated by the underscore feature. All the remaining lines between entries were generated in this fashion.

The commands in lines 110 to 130 changed the print style. The loop in lines 140 to 210 generated the body of the table.

Notice that the LPRINT command in line 200 sent the same data to the printer each time that it was executed. But, the statement in line 190 changed the international character set each time through the loop. So, the data recevied from the LPRINT in line 200 was translated to a different international character set each time.

FX SERIES PROGRAM

The program to generate the table using an FX Series program follows:

```
10 MODE=24:GOSUB 370
20 LPRINT CHR$(27)"-1";
30 GOSUB 320
40 LPRINT CHR$(14);
50 FOR N=1 TO 2
60 READ A$
70 LPRINT A$;
80 LPRINT CHR$(27)"4";
90 NEXT N
100 LPRINT CHR$(27)"5"
110 MODE=0:GOSUB 370
```

```
120 FOR J=0 TO 8
130 READ A$
140 LPRINT CHR$(27)"E";
150 LPRINT A$;
160 LPRINT CHR$(27)"F";
170 LPRINT CHR$(27)"R"CHR$(J);
180 LPRINT " # $ @ [ \ ] ^ ` { | } ~"
190 NEXT J
200 LPRINT CHR$(27)"@"
210 END
220 DATA "COUNTRY ","b                    "
230 DATA "   USA                0      "
240 DATA "   FRANCE             1      "
250 DATA "   GERMANY            2      "
260 DATA "   ENGLAND            3      "
270 DATA "   DENMARK I          4      "
280 DATA "   SWEDEN             5      "
290 DATA "   ITALY              6      "
300 DATA "   SPAIN              7      "
310 DATA "   JAPAN              8      "
320 FOR I=1 TO 46
330 LPRINT " ";
340 NEXT I
350 LPRINT
360 RETURN
370 LPRINT CHR$(27)"!"CHR$(MODE);
380 RETURN
```

The program output is:

COUNTRY	**b**												
USA	0	#	$	@	[	\	]	^	`	{	¦	}	~
FRANCE	1	#	$	à	°	ç	§	^	`	é	ù	è	¨
GERMANY	2	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
ENGLAND	3	£	$	@	[	\	]	^	`	{	¦	}	~
DENMARK I	4	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
SWEDEN	5	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
ITALY	6	#	$	@	°	\	é	^	ù	à	ò	è	ì
SPAIN	7	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
JAPAN	8	#	$	@	[	¥	]	^	`	{	¦	}	~

This table has two fewer entries than table 8.1 because the FX Series has nine international characters sets instead of the 11 that are available on the RX Series. Also, the columns spacing is different because the longer entries in the first column were eliminated. The necessary spaces to accomplish horizontal formatting were inserted in the DATA statements. The horizontal tab was not used because the underline feature was activated to generate the lines between entries. Output was generated in every column of the table. So, none of the columns could be skipped.

The statements in lines 10 and 110 changed the print mode with the Master Select Mode. The Master Select Mode appeared in the subroutine. The command in line 20 enabled the underlined mode. The subroutine called in line 30 generated the top line of the table.

The loop in lines 50 to 90 caused the column headings to be printed. The loop in lines 120 to 190 generated the body of the table. Notice that the LPRINT command in line 180 sent the same data to the printer each time that it was executed. But, the statement in line 170 changed the international character set each time through the loop. So the data received from the LPRINT in line 180 was translated to a different international character set each time.

Creating a Bar Graph

Printers can be used to generate different types of graphs. In this example program we show how to create a bar graph.

The first step is to decide upon the horizontal format. We decided on a maximum width of 4⅝ inches so that the table would fit across one page of this book. Next, we made a crude sketch of the desired table. Figure 12.1 shows what the drawing looked like after we were completely finished planning the table. We decided to use the compressed pitch mode in this example. Room exists for 79 compressed characters in a width of 4⅝ inches. The numbers at the bottom of the sketch indicate at which columns we wanted to place objects on the bar graph. We used these values to choose the horizontal tab settings.

We then had to determine the vertical formatting. We decided to print the chart using a line spacing of one dot. The figures at the upper right show that we decided upon a vertical spacing of approximately 5/16

inch for ten units on the graph. A space of 5/16 inch turned out to be close to 23, one dot = 1/72 inch line. We used this value to control the loops that generated the table. With this information in hand, we developed the program. The graph was changed slightly from the original plan when we modified the FX program to operate on an older RX printer.

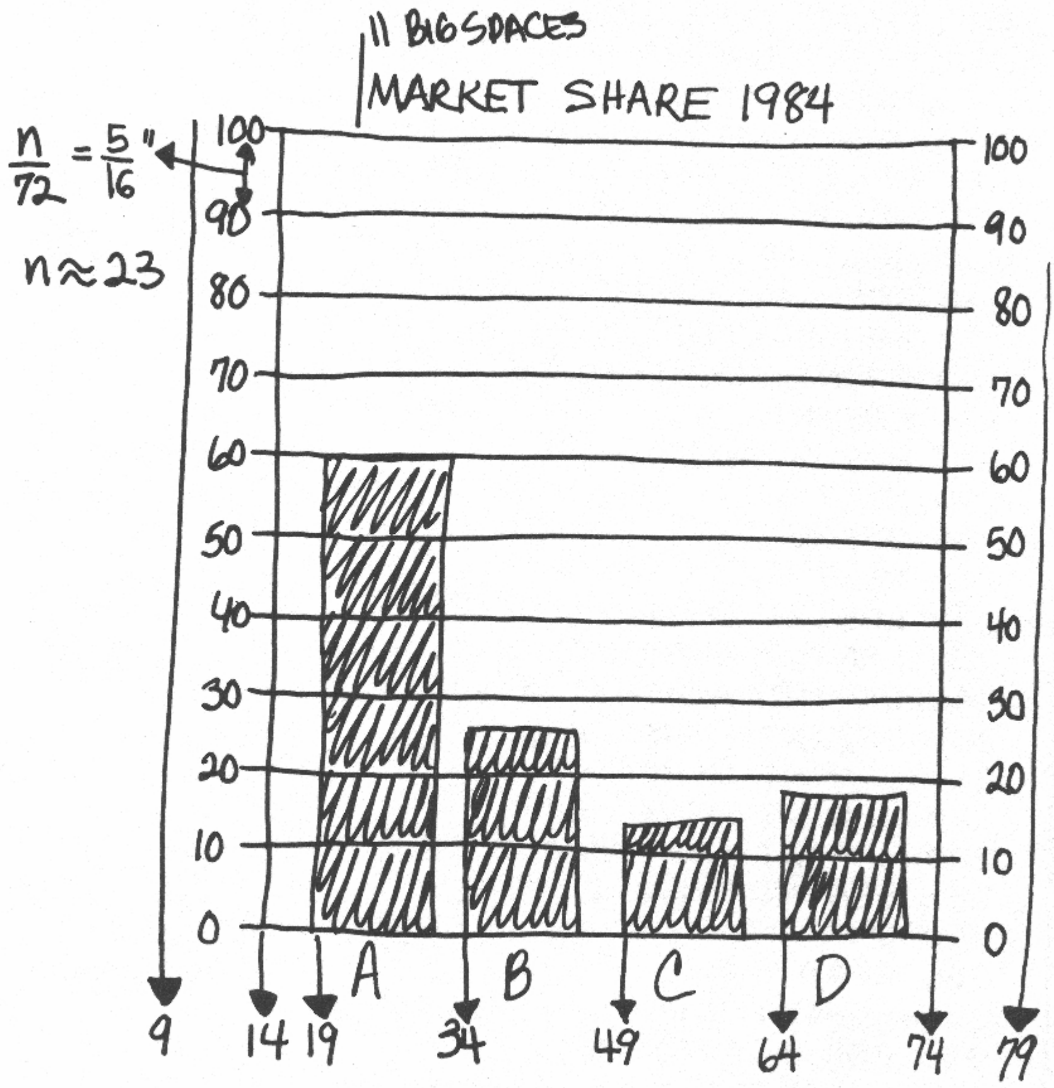


FIGURE 12.1. Sketch used to plan FX bar chart

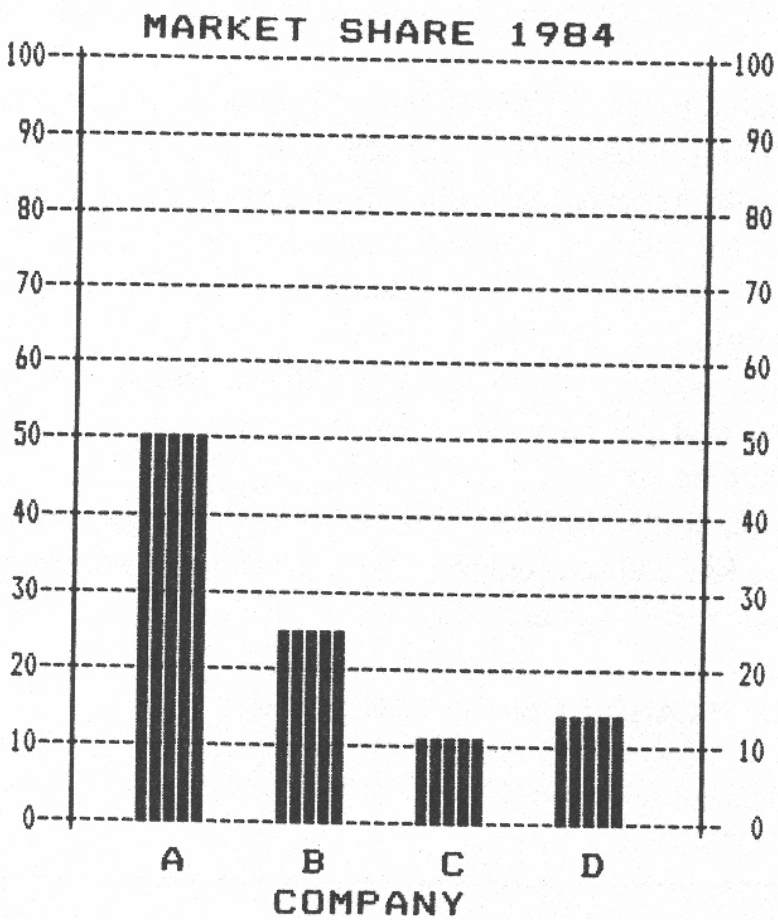
OLDER RX SERIES PROGRAM

The program to generate a bar graph using an older RX Series printer is:

```
10 H$=CHR$(9)
20 PRCNT=100
30 DIM MAX(4)
40 LPRINT CHR$(15);
50 LPRINT CHR$(27)"e0"CHR$(5);
60 LPRINT CHR$(14);
70 LPRINT H$H$H$"MARKET SHARE 1984"
110 LPRINT CHR$(27)"A"CHR$(1);
120 GOSUB 320
130 READ MAX(1),MAX(2),MAX(3),MAX(4)
140 FOR J=1 TO 10
150 FOR K=1 TO 23
160 PRCNT=PRCNT-10/23
170 LPRINT
180 GOSUB 390
190 NEXT K
200 GOSUB 320
210 NEXT J
220 LPRINT CHR$(27)"2"
230 LPRINT CHR$(27)"W1";
240 LPRINT H$H$H$" A";
250 LPRINT H$H$" B";
260 LPRINT H$H$" C";
270 LPRINT H$H$" D"
280 LPRINT H$H$H$H$H$"COMPANY"
281 LPRINT CHR$(27)"@"
290 END
320 READ NUM$
330 LPRINT CHR$(13);H$;NUM$;
340 FOR I=1 TO 49
350 LPRINT "-";
360 NEXT I
370 LPRINT NUM$;
380 RETURN
390 LPRINT H$H$"!";
400 FOR L=1 TO 4
410 LPRINT H$;
```

```
420 IF MAX(L)<PRCNT THEN LPRINT "      ";
                    ELSE LPRINT "------";
430 NEXT L
440 LPRINT H$"|";
450 RETURN
460 DATA 100,50,25,11,14," 90"," 80"
470 DATA " 70"," 60"," 50"," 40"," 30"
480 DATA " 20"," 10","  0"
```

The program output is:



The statement in line 10 defined "H$" to be the BASIC expression for a horizontal tab. This action was taken to make the program easier to read and to decrease the number of keystrokes needed to enter the program. The statement in line 20 initialized the counter. The counter was used to

calculate the value of the market percentage for each increment on the graph. The statement in lines 40 and 60 selected the print mode. The command in line 50 set the tab stop interval. The statement in line 70 printed the title of the graph. Note that each appearance of an "H$" caused a horizontal tab.

The statement in line 110 caused the line spacing to be changed to 1/72 of an inch. This spacing is equivalent to one dot. The subroutine called in line 120 caused the current value of the market share percentage and the dotted line across the graph to be printed.

The READ statement in line 130 obtained the value of the percent market share for each of the four companies. Note that these are the second through fourth values that appear in the DATA statement in line 460.

The nested loops in line 140 to 210 generated the body of the table. The inner loop created each of the ten units. The outer loop caused the inner loop tobe executed ten times. Note that the statement in line 160 incremented the counter. The increment was 10/23. Each interval of ten percentage points used 23 lines. Each line was equivalent to 10/23 of a percentage point.

The subroutine that started line 390 generated the right and left vertical lines and the bars. The IF statement in line 420 controlled the height of each bar.

The command in line 220 returned the line spacing to 1/6 inch. The command in line 230 selected the enlarged mode. The statements in lines 240 to 280 caused the graph labels to be printed.

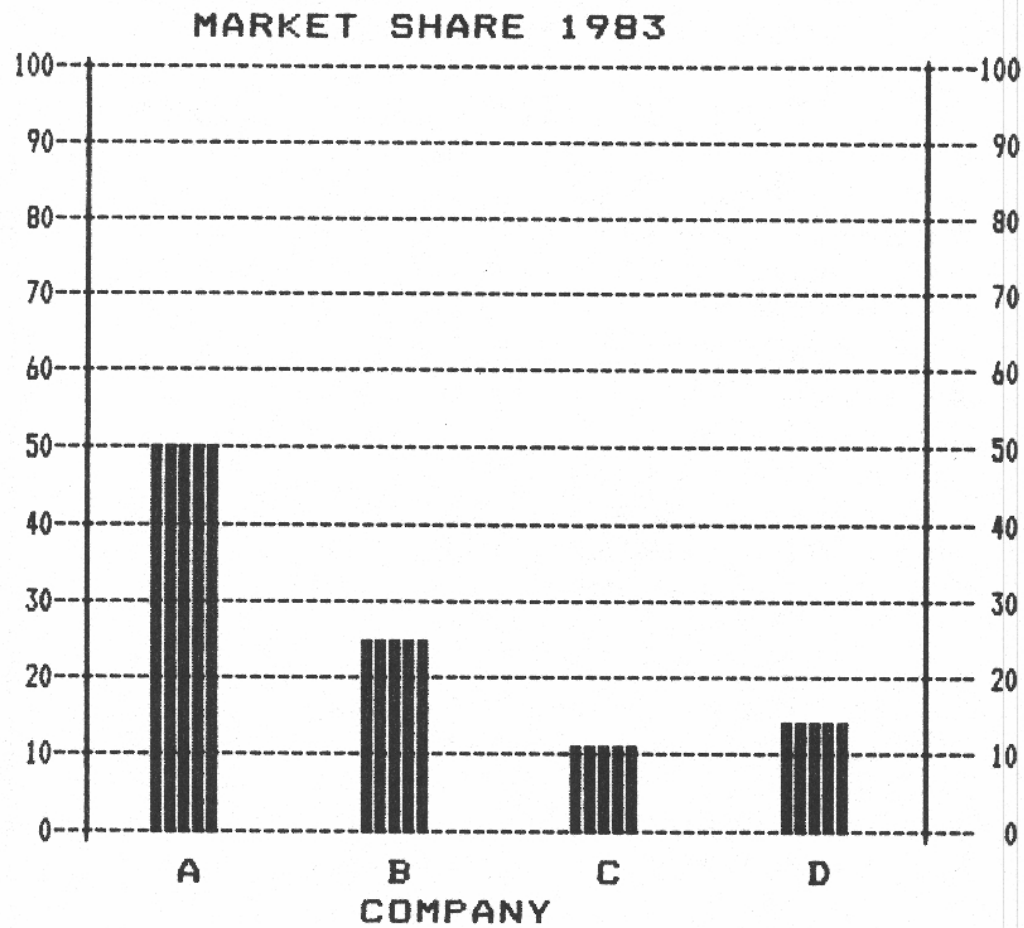
NEWER RX AND FX SERIES PROGRAM

The program to generate a bar graph using a newer RX or FX Series printer follows:

```
10 H$=CHR$(9)
20 PRCNT=100
30 DIM MAX(4)
40 LPRINT CHR$(15)CHR$(14);
50 LPRINT CHR$(27)"D"CHR$(11)CHR$(1);
60 LPRINT H$"MARKET SHARE 1983"
```

```
80 LPRINT CHR$(27)"D"CHR$(9)CHR$(14);
90 LPRINT CHR$(19)CHR$(34)CHR$(49);
100 LPRINT CHR$(64)CHR$(74)CHR$(1);
110 LPRINT CHR$(27)"A"CHR$(1);
120 GOSUB 330
130 READ MAX(1),MAX(2),MAX(3),MAX(4)
140 FOR J=1 TO 10
150 FOR K=1 TO 23
160 PRCNT=PRCNT-10/23
170 LPRINT
180 GOSUB 400
190 NEXT K
200 GOSUB 330
210 NEXT J
220 LPRINT CHR$(27)"2"
230 LPRINT CHR$(27)"W1";
240 LPRINT H$H$H$" A";
250 LPRINT H$" B";
260 LPRINT H$" C";
270 LPRINT H$" D"
280 LPRINT H$H$H$H$"COMPANY"
290 LPRINT CHR$(27)"@"
300 END
330 READ NUM$
340 LPRINT CHR$(13);H$;NUM$;
350 FOR I=1 TO 66
360 LPRINT "-";
370 NEXT I
380 LPRINT NUM$;
390 RETURN
400 LPRINT H$H$"|";
410 FOR L=1 TO 4
420 LPRINT H$;
430 IF MAX(L)<PRCNT THEN LPRINT "     ";
                    ELSE LPRINT "-----";
440 NEXT L
450 LPRINT H$"|";
460 RETURN
470 DATA 100,50,25,11,14," 90"," 80"
480 DATA " 70"," 60"," 50"," 40"," 30"
490 DATA " 20"," 10","  0"
```

The program output is:



The statement in line 10 defined "H$" to be the BASIC expression for a horizontal tab. This action was taken to make the program easier to read and to decrease the number of keystrokes needed to enter the program. The statement in line 20 initialized the counter. The counter was used to calculate the value of the market share percentage for each increment in the graph. The statement in line 40 selected the print mode. The statement in line 50 set the horizontal tab stop for the title line. The command in line 60 caused the chart title to be printed.

The statements in lines 80 to 100 set the horizontal tab stops for the chart. The command in line 110 caused the line spacing to be changed to 1/72 of an inch. This spacing is equivalent to one dot. The subroutine called in line 120 caused the current value of the market share percentage to be printed along with the dotted line across the graph.

The READ statement in line 130 obtained the value of the percent market share for each of the four companies. Note that these are the second through fourth values that appear in the DATA statement in line 470.

The nested loop in lines 140 to 210 generated the body of the table. The inner loop created each ten unit interval. The outer loop caused the inner loop to be executed ten times. Note that the statement in line 160 incremented the counter. The increment was 10/23. Each interval of ten percentage points used 23 lines. Each line was equivalent to 10/23 of a percentage point.

The subroutine that started at line 400 generated the right and left vertical lines and the bars. The IF statement in line 430 controlled the height of each bar.

The command in line 220 returned the line spacing to 1/6 inch. The statements in lines 240 to 280 caused the graph labels to be printed.

Handling Single-Sheet Paper (Newer RX and FX Printers Only)

The Newer RX and FX Series printers can accommodate single-sheet typing paper. The example program shows how to generate letters with different vertical formats. The vertical tab stops for each letter are saved in channels.

In this example we assumed that we had two types of form letters that could be sent. We used a different vertical format for each letter. The program follows:

```
10 GOSUB 420
20 LPRINT CHR$(27)"@";
30 LPRINT CHR$(27)"8";
40 V$=CHR$(11):H$=CHR$(9)
50 LPRINT CHR$(27)"M";
60 LPRINT CHR$(27)"l"CHR$(12);
70 LPRINT CHR$(27)"Q"CHR$(84);
80 LPRINT CHR$(27)"b"CHR$(1)CHR$(8)CHR$(12);
90 LPRINT CHR$(18)CHR$(20)CHR$(54);
100 LPRINT CHR$(58)CHR$(1);
110 LPRINT CHR$(27)"b"CHR$(2)CHR$(20)CHR$(25);
```

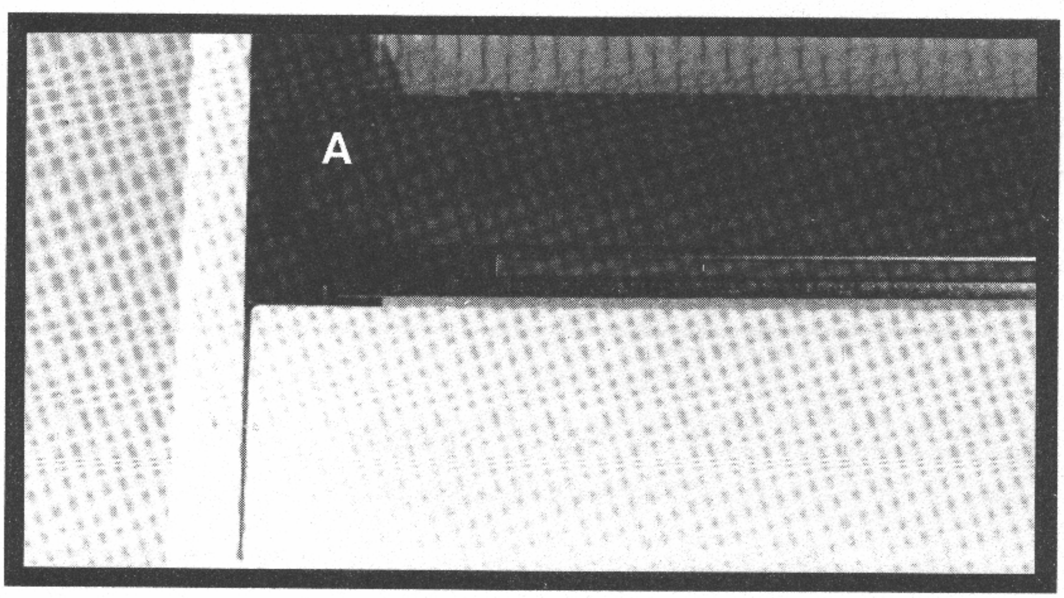
```
120 LPRINT CHR$(31)CHR$(33)CHR$(42)CHR$(46);
130 LPRINT CHR$(1);
140 LPRINT CHR$(27)"D"CHR$(5)CHR$(50)CHR$(1);
150 INPUT "TYPE OF LETTER--1 OR 2";TYPE
160 LPRINT CHR$(27)"/"CHR$(TYPE);
170 LPRINT V$;
180 LPRINT H$H$"9754 Elbur Way"
190 LPRINT H$H$"Cleveland, Ohio 44126"
200 INPUT "MONTH,DAY,YEAR";MO$,D$,YR$
210 LPRINT H$H$MO$" "D$", "YR$
220 LPRINT V$;
230 INPUT "1) NAME
           2) NUMBER AND STREET
           3) CITY STATE AND ZIP";
NM$,NMBR$,CSZ$
240 LPRINT NM$
250 LPRINT NMBR$
260 LPRINT CSZ$
270 LPRINT V$;
280 LPRINT "DEAR "NM$":"
290 GOSUB 370
300 LPRINT H$H$"Sincerely,"
310 LPRINT V$;
320 LPRINT H$H$"Thomas Streets"
330 INPUT "ANOTHER--Y OR N";Y$
340 IF Y$="Y" THEN GOSUB 420
350 LPRINT CHR$(27)"@"
360 END
370 LPRINT
380 LPRINT H$"NORMALLY THE TEXT OF THE"
390 LPRINT H$"LETTER WOULD APPEAR HERE"
400 LPRINT V$;
410 RETURN
420 LPRINT CHR$(7);
430 PRINT "PUT PAPER IN PRINTER"
440 PRINT "POSITION THE TOP EDGE AT"
450 PRINT "THE PRINT HEAD"
460 INPUT "STRIKE RETURN KEY WHEN READY";Z$
470 RETURN 20
```

When the program is run, the user must perform several actions and answer prompts as they appear on the screen. The first message that appears on the screen follows:

```
PUT PAPER IN PRINTER
POSITION THE TOP EDGE AT
THE PRINT HEAD
STRIKE RETURN KEY WHEN READY
```

To accomplish the loading of the printer with single-sheet paper, perform the following steps:

1. Remove any paper presently in the printer.
2. Remove the paper guide and both dust covers.
3. Move the two pin feed units to their most extreme outboard position.
4. Pull the paper bail toward the front of the printer.
5. Verify that the paper release lever is set toward the rear of the printer. Figure 12.2 shows the location of the release lever. The release lever controls the friction feed mechanism.



a, Release lever

FIGURE 12.2. Locating the release lever

6. Load the paper into the printer the same as you would into a typewriter. The scale on the paper bail can be used to center the paper in the printer.
7. Use the manual feed knob to position the top edge of the paper even with the print head.

The paper is now positioned. The user should now strike the return key. The user should be alert when the printer starts moving the paper. We found that it is necessary to help guide the top edge of the paper. If this precaution is not taken, the top edge of the paper will catch on the paper bail causing the printer to jam. Once paper movement has stopped, the bail should be lowered.

After the user strikes the return key, the screen will display:

```
TYPE OF LETTER -- 1 OR 2?
```

Typing "1" and depressing the return key selects the longer letter. A "2" will cause the shorter letter to be selected. Paper movement will begin immediately after the return key is depressed.

Next, the screen will display:

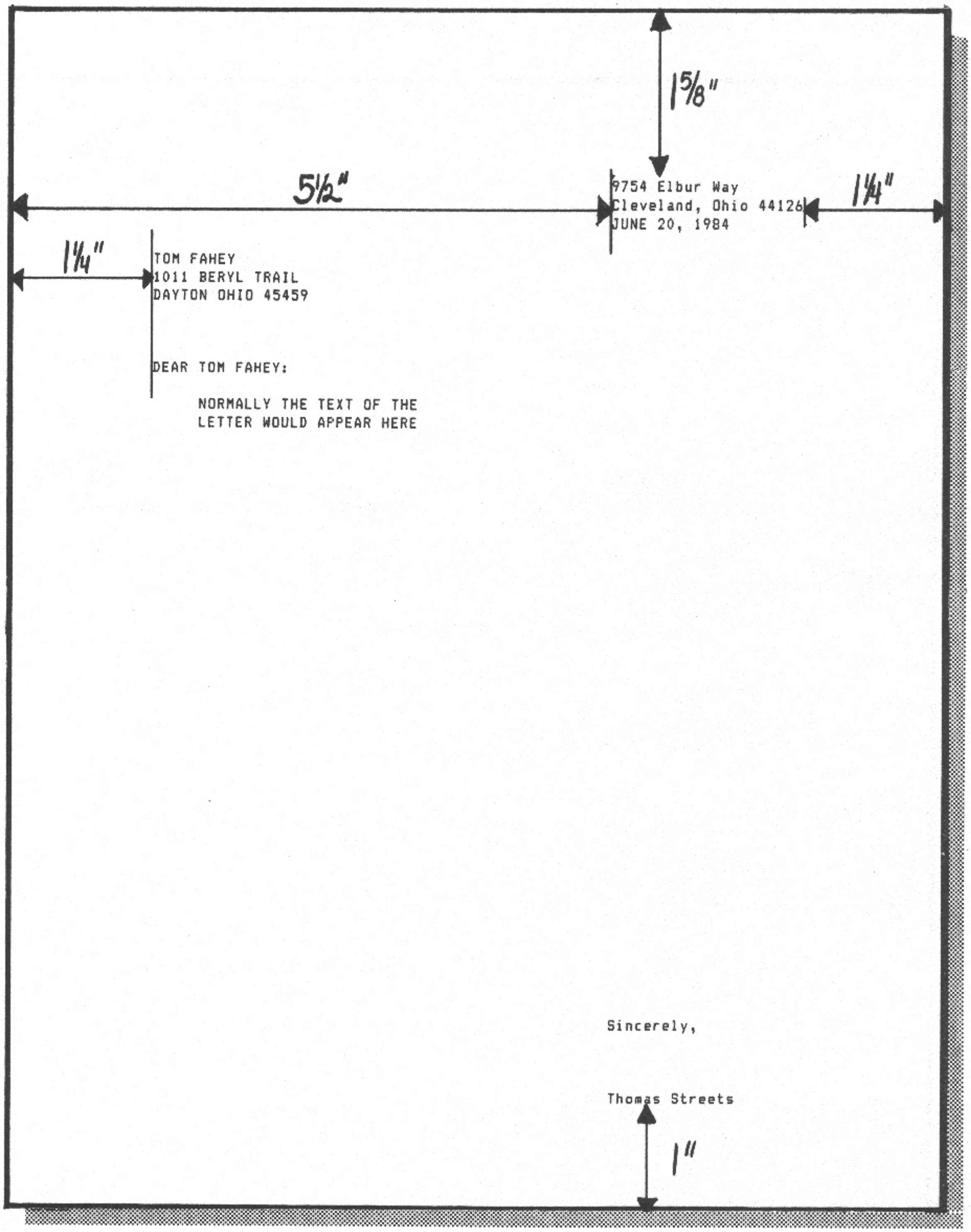
```
MONTH, DAY, YEAR?
```

We typed: "May, 21, 1984" and pressed the return key. The following prompt then appeared on the screen:

```
1. NAME
2. NUMBER OF STREET
3. CITY STATE AND ZIP
```

We typed: "Mr. Tom Fahey, 1011 Beryl Trail, Cleveland Ohio 44102" and pressed the return key.

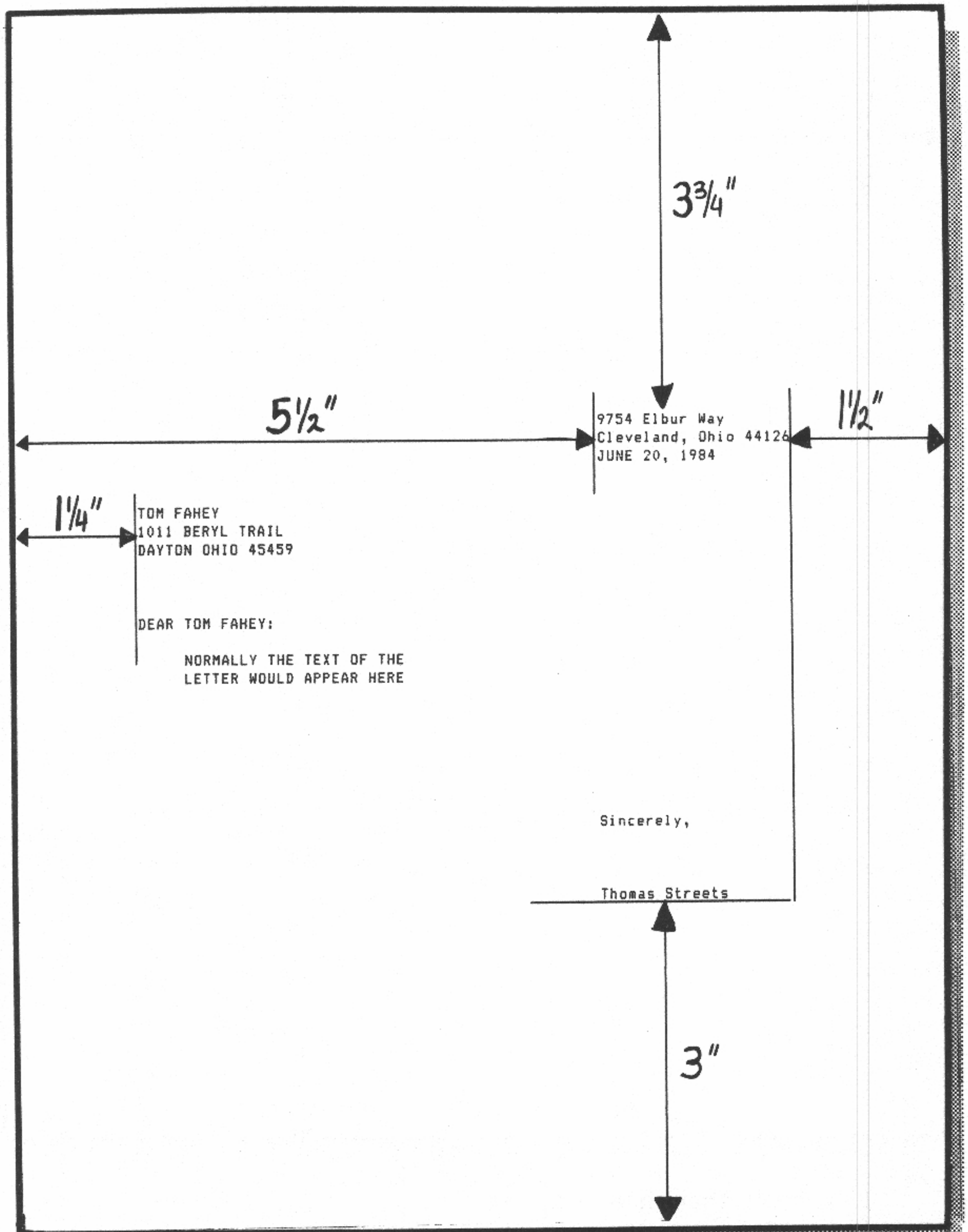
The final output was formatted as follows:



The screen then displayed:

ANOTHER -- Y or N?

We typed "Y" and pressed the return key. This time we chose the option for the shorter letter. We answered the other prompts as we had for the first letter. This time, the final output was formatted as follows:



When the option:

ANOTHER -- Y or N?

appeared on the screen, we chose to exit from the program by typing "N".

The subroutine called in line 10 caused the message to load the paper to be displayed on the screen. The command in line 20 sent the Master Reset Code. This action was important because it marked the top-of-form position correctly.

The statement in line 30 deactivated the paper out sensor. The commands in lines 60 and 70 caused the right and left margin settings to be changed. Notice that the print pitch was determined by the command in line 50 before the margins were set.

The statements in lines 80 to 130 caused two sets of vertical tab stops to be stored in separate channels. The command in line 160 caused one of the channels to be selected. Note that the channel to be selected was interactively specified by the user in line 150.

Filling Out a Preprinted Form

Printers can generally be used to fill out preprinted forms. In this example we show how Epson printers can be used for this important application.

The first step is to measure the form. Both the vertical and horizontal position of the blanks to be filled must be determined. Figure 12.3 shows a copy of the form that we are using in this example. Note that we marked and measured the approximate position of each blank that needs to be filled.

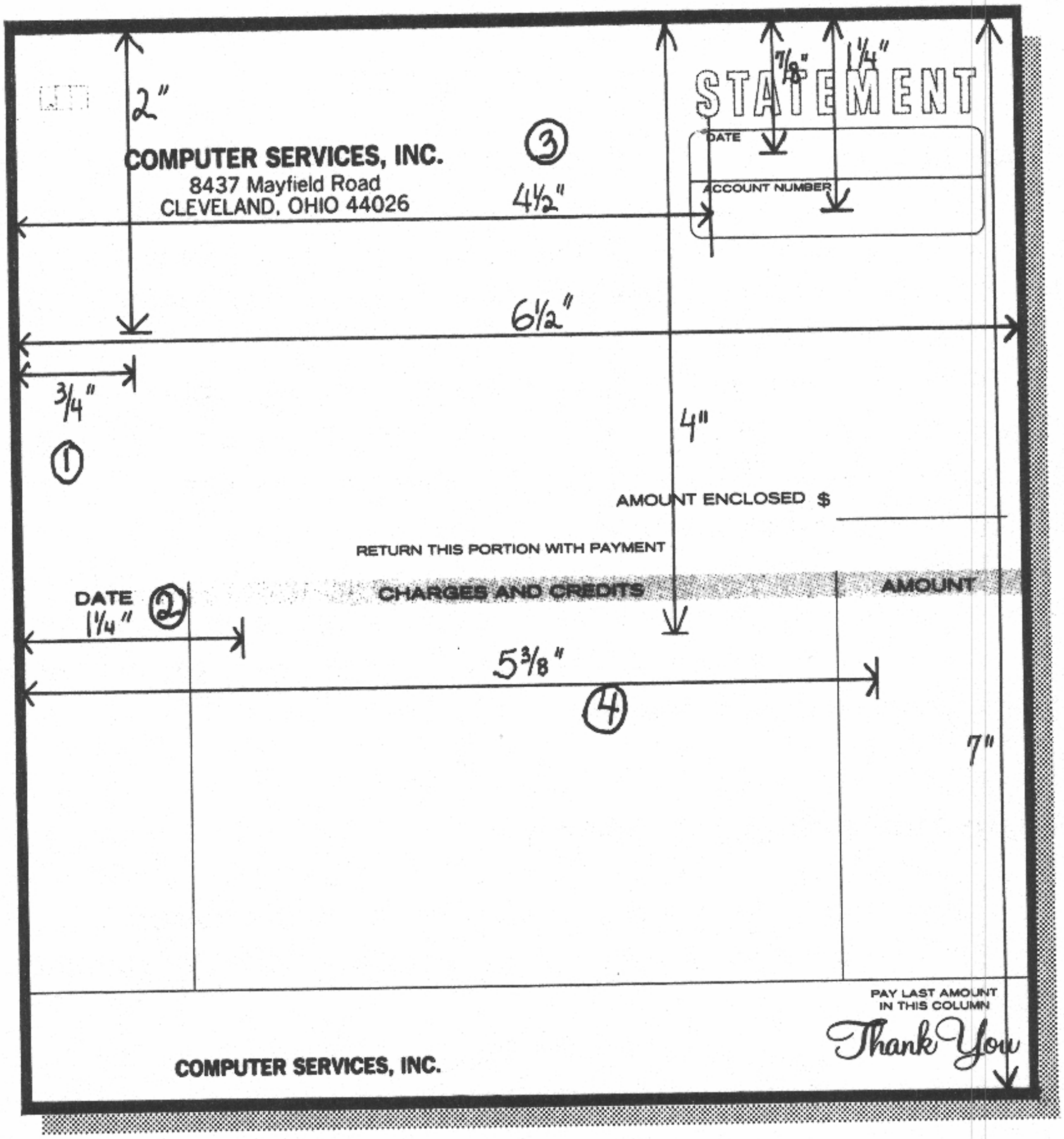


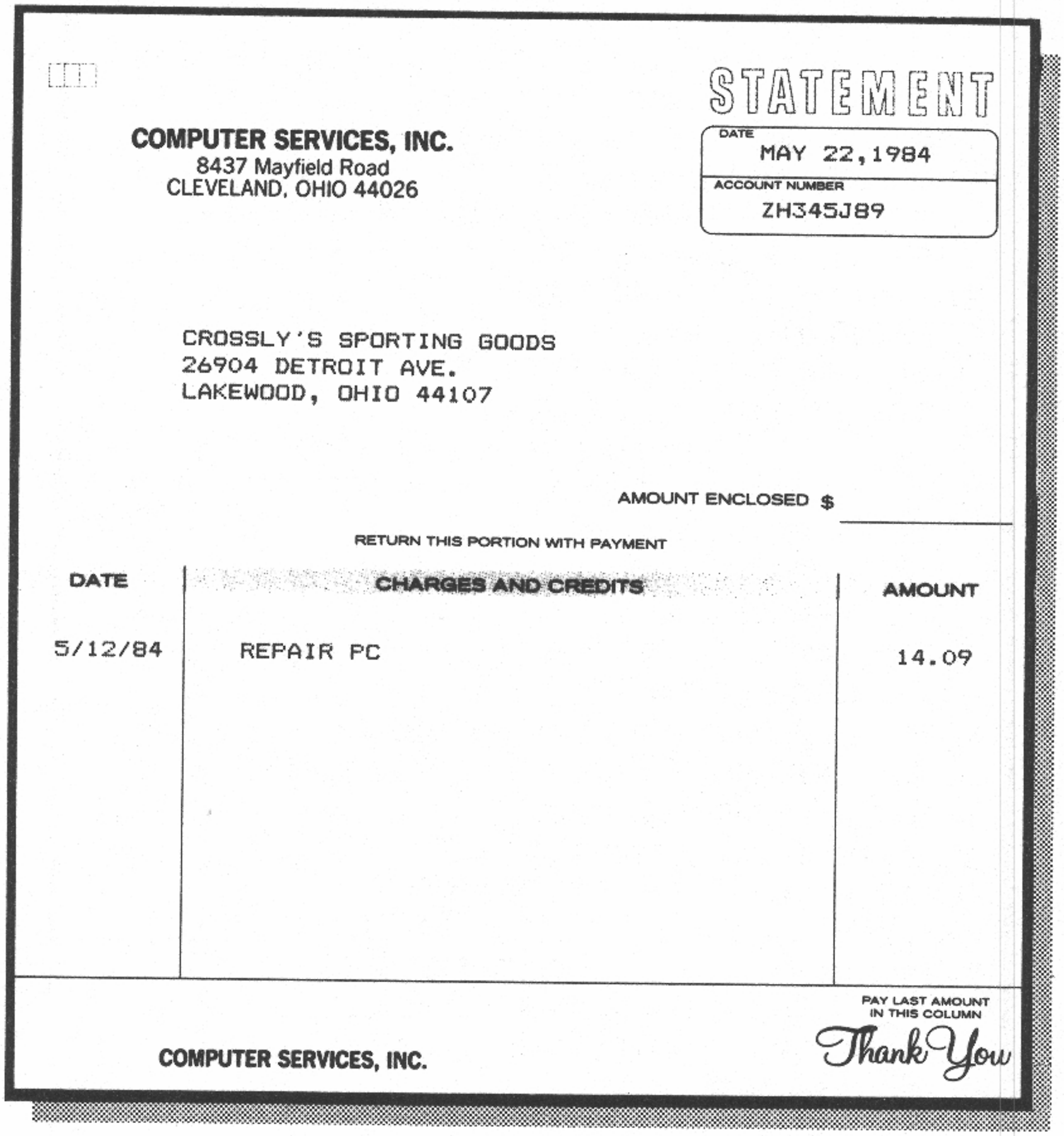
FIGURE 12.3. The marked form. The form was reduced 65%.

OLDER RX SERIES PROGRAM

The program to fill out the preprinted form using an older RX Series printer follows:

```
10 H$=CHR$(9):V$=CHR$(11)
20 LPRINT CHR$(27)"C"CHR$(0)CHR$(7);
30 LPRINT CHR$(27)"0";
40 LPRINT CHR$(27)"e1"CHR$(3);
50 LPRINT CHR$(27)"2";
60 LPRINT CHR$(27)"e0"CHR$(21);
70 GOSUB 250
80 LPRINT V$V$H$H$"   ";
90 LPRINT CURDT$
100 LPRINT V$H$H$"   ";
110 LPRINT ACCNMBR$
120 LPRINT CHR$(27)"e1"CHR$(12);
130 LPRINT CHR$(27)"e0"CHR$(4);
140 LPRINT V$H$;
150 LPRINT NM$
160 LPRINT H$STRT$
170 LPRINT H$CTY$
180 LPRINT V$;
190 LPRINT WRKDT$H$H$WRK$;
200 LPRINT CHR$(27)"e0"CHR$(18);
210 LPRINT H$H$AMNT$
220 LPRINT CHR$(12);
230 LPRINT CHR$(27)"@";
240 END
250 READ CURDT$,ACCNMBR$
260 READ NM$,STRT$,CTY$
270 READ WRKDT$,WRK$,AMNT$
280 RETURN
290 DATA "MAY 22,1984",ZH345J89
300 DATA "CROSSLY'S SPORTING GOODS"
310 DATA "26904 DETROIT AVE."
320 DATA "LAKEWOOD, OHIO 44107"
330 DATA "5/12/84"
340 DATA "REPAIR PC"
350 DATA "14.09"
```

The program output is:



STATEMENT

COMPUTER SERVICES, INC.
8437 Mayfield Road
CLEVELAND, OHIO 44026

DATE
MAY 22,1984

ACCOUNT NUMBER
ZH345J89

CROSSLY'S SPORTING GOODS
26904 DETROIT AVE.
LAKEWOOD, OHIO 44107

AMOUNT ENCLOSED $

RETURN THIS PORTION WITH PAYMENT

DATE	CHARGES AND CREDITS	AMOUNT
5/12/84	REPAIR PC	14.09

PAY LAST AMOUNT IN THIS COLUMN

COMPUTER SERVICES, INC.

Thank You

Note that the form has been reduced by 65%.

Before the program was executed, the top-of-form position was correctly set. The command in line 20 caused the form length to be changed to seven inches.

The statement in line 30 caused the line spacing to be changed to 1/8 inch. The command in line 40 generated vertical tab stops at 3/8 inch intervals down the page. The statement in line 50 returned the line spacing to 1/6 inch. The command in line 60 generated horizontal tab stops at 2-1/10 inch intervals across the page.

The subroutine called in line 70 used the READ command to obtain the necessary data values. In a real application, the subroutine would probably be set up to read from a separate file. Since we only wanted to demonstrate the spacing, we stuck with only one set of data.

The statements in line 80 to 110 used vertical and horizontal tabs to position the output on the page. For example, the statement in line 80 caused two vertical tabs. These tabs caused the paper to be moved 6/8 inch. This statement also caused two horizontal tabs. These tabs caused the next print position to be changed to column 42 (4-2/10 inches) across the page. The two spaces in the LPRINT command in line 80 were used to "fine tune" the horizontal placement of the output. The statement in line 90 caused the current date to be output.

The statements in lines 120 and 130 caused the vertical and horizontal tabs to be redefined. Redefining the tab stops allowed the output to be formatted more easily. For example, the new vertical tab stops were defined every two inches down the page. So, only one vertical tab must be executed to reach the vertical position of the site where the address will be output. Using the previous vertical tab stop settings, we would have needed several vertical tabs to accomplish the same positioning.

The remainder of the program functions in a fashion similar to the portion already explained.

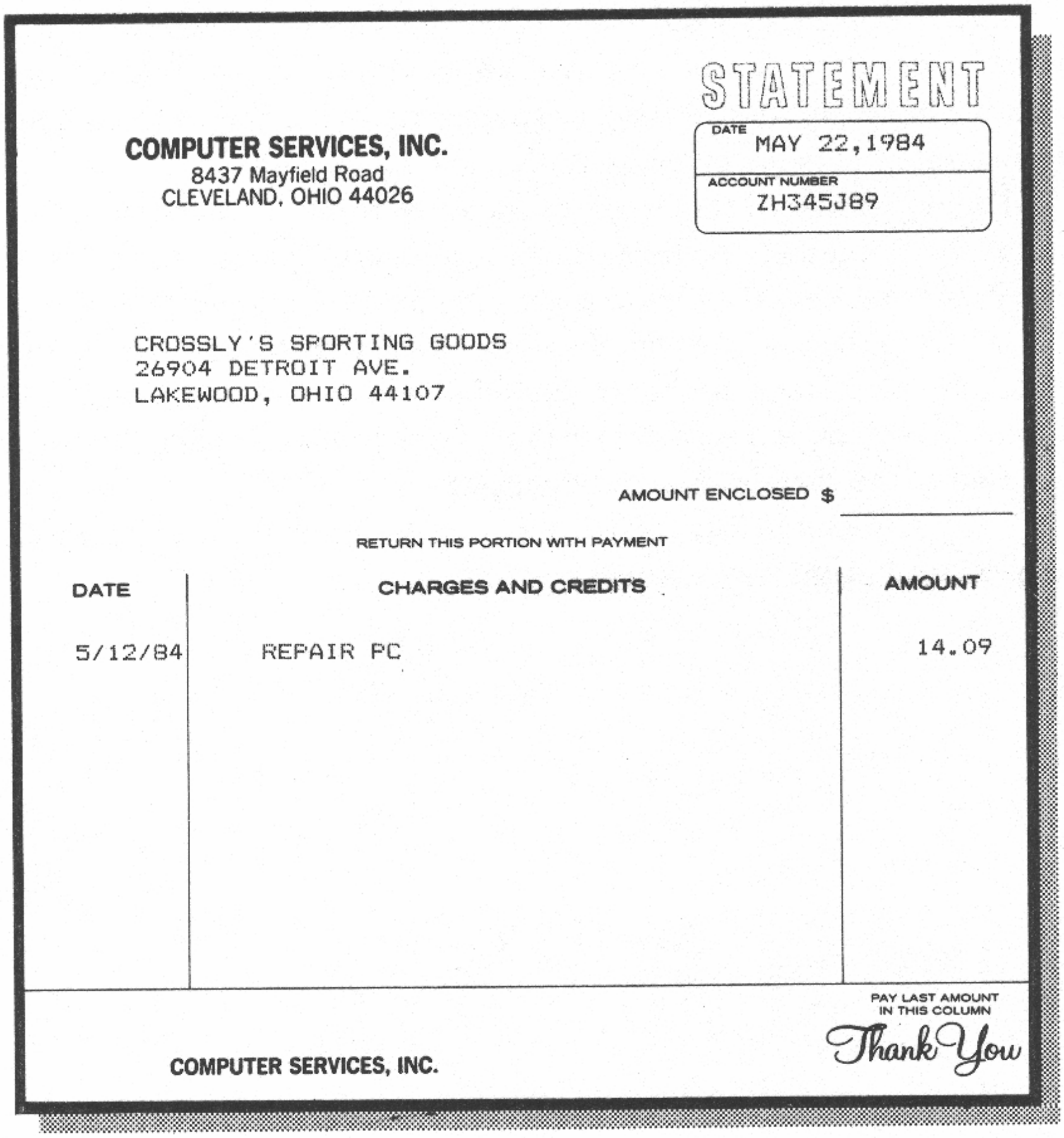
The command in line 220 is a form feed. This command positioned the print head at the top-of-form position.

NEWER RX AND FX SERIES PROGRAM

The program to fill out the preprinted form using a newer RX Series or an FX series printer follows:

```
10 H$=CHR$(9):V$=CHR$(11)
20 LPRINT CHR$(27)"C"CHR$(0)CHR$(7);
30 LPRINT CHR$(27)"O";
40 LPRINT CHR$(27)"B"CHR$(6)CHR$(9);
50 LPRINT CHR$(16)CHR$(32)CHR$(1);
60 LPRINT CHR$(27)"2";
70 LPRINT CHR$(27)"D"CHR$(8)CHR$(12);
80 LPRINT CHR$(45)CHR$(54)CHR$(1);
90 GOSUB 1000
100 LPRINT V$H$H$H$;
110 LPRINT CURDT$
120 LPRINT V$H$H$H$;
130 LPRINT ACCNMBR$
140 LPRINT V$H$;
150 LPRINT NM$
160 LPRINT H$STRT$
170 LPRINT H$CTY$
180 LPRINT V$;
190 LPRINT WRKDT$H$H$WRK$H$H$AMNT$
200 LPRINT CHR$(12);
210 LPRINT CHR$(27)"@";
220 END
1000 READ CURDT$,ACCNMBR$
1010 READ NM$,STRT$,CTY$
1020 READ WRKDT$,WRK$,AMNT$
1030 RETURN
1040 DATA "MAY 22,1984",ZH345J89
1050 DATA "CROSSLY'S SPORTING GOODS"
1060 DATA "26904 DETROIT AVE."
1070 DATA "LAKEWOOD, OHIO 44107"
1080 DATA "5/12/84"
1090 DATA "REPAIR PC"
1100 DATA "14.09"
```

The program output is:



STATEMENT

COMPUTER SERVICES, INC.
8437 Mayfield Road
CLEVELAND, OHIO 44026

DATE
MAY 22,1984

ACCOUNT NUMBER
ZH345J89

CROSSLY'S SPORTING GOODS
26904 DETROIT AVE.
LAKEWOOD, OHIO 44107

AMOUNT ENCLOSED $

RETURN THIS PORTION WITH PAYMENT

DATE	CHARGES AND CREDITS	AMOUNT
5/12/84	REPAIR PC	14.09

PAY LAST AMOUNT IN THIS COLUMN

COMPUTER SERVICES, INC.

Thank You

Note that the form has been reduced by 65%.

Before the program was executed, the top-of-form position was set correctly. The command in line 20 caused the form length to be changed to seven inches.

The statement in line 30 caused the line spacing to be changed to 1/8 inch. The statements in lines 40 and 50 generated vertical tab stops at the sites determined in figure 12.3 on the marked form. The command in line 60 returned the line spacing to 1/6 inch. The statement in lines 70 and 80

generated horizontal tab stops at the sites determined on the marked form.

The subroutine called in line 90 used the READ command to obtain the necessary data values. In a real application, the subroutine would probably be set up to read from a separate file. Since we only wanted to demonstrate the spacing, we used only one set of data.

The remainder of the program used horizontal and vertical tabs to format the output on the form. For example, the statement in line 100 caused one vertical tab. This vertical tab caused the paper to be moved 6/8 inch from the top-of-form position. This statement also caused three horizontal tabs. These horizontal tabs changed the next print position to column 45 (4½ inches) across the paper. The command in line 110 caused the value of the current date to be output

The command in line 200 is a form feed. This command positioned the printhead at the top-of-form position.

13

Selecting Modes Through DIP Switches

Introduction

The RX and FX Series printers both have DIP (Dual In-line Package) switches incorporated into their hardware. Figure 13.1 shows a typical DIP switch. Notice that a DIP switch is made up of individual pins. Each pin can be set either ON or OFF.

The pin settings determine the default conditions. Default conditions specify how the printer will operate when the power is turned ON. Generally, each pin controls a printer function. A printer function is a mode or a mechanical action. An example of a mechanical action is sounding the buzzer. Table 13.1 lists the ten functions controlled by the DIP switches and the default conditions set by the factory.

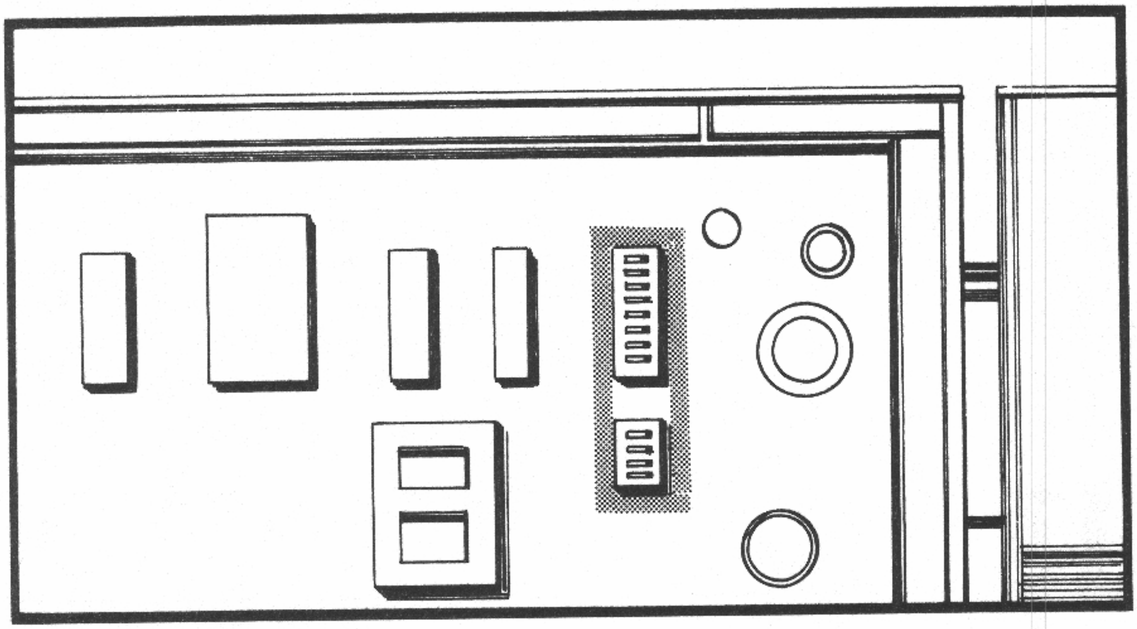


FIGURE 13.1. A typical DIP switch

Table 13.1. Printer functions controlled by the DIP switches

Printer Function	Factory Set Default Conditions
Print Mode	Pica-size characters
Graphic Symbol Mode (RX only)	Graphic symbols not selected
Buzzer	Buzzer will sound
Form Length (RX only)	11 inch form length
Paper Out Sensor	Sensor active
International Character Set	USA character set selected
Zero Font	Prints zero as "0"
Printer Select	Printer always active
Automatic Line Feed	No line feed added to carriage return
Perforation Skip-over	Skip-over not active
RAM memory (FX only)	Memory available for user defined characters
Compressed Mode (FX only)	Pica-size characters

If you find yourself often using control codes to change a particular default condition, you can instead change the DIP switch setting. The printer will then default to this often used condition. It will no longer be necessary to constantly send the same control code to change modes each time you use the printer.

For example, if you generally want to skip the perforation, change the pin that controls the skip-over function. The printer will now automatically skip the perforations. This change will relieve you of the burden of sending the ESC N code each time you use the printer.

Method of Setting Any DIP Switch

All electronic hardware should be treated gently. DIP switches are no exception to this rule. Avoid touching the switches with your hands. Static charge, moisture, and body oils from your fingers are all harmful to the switches. Be sure the power is turned OFF before changing any pin setting. Use the tip of a small screwdriver to change the DIP switch pin from one position to the other.

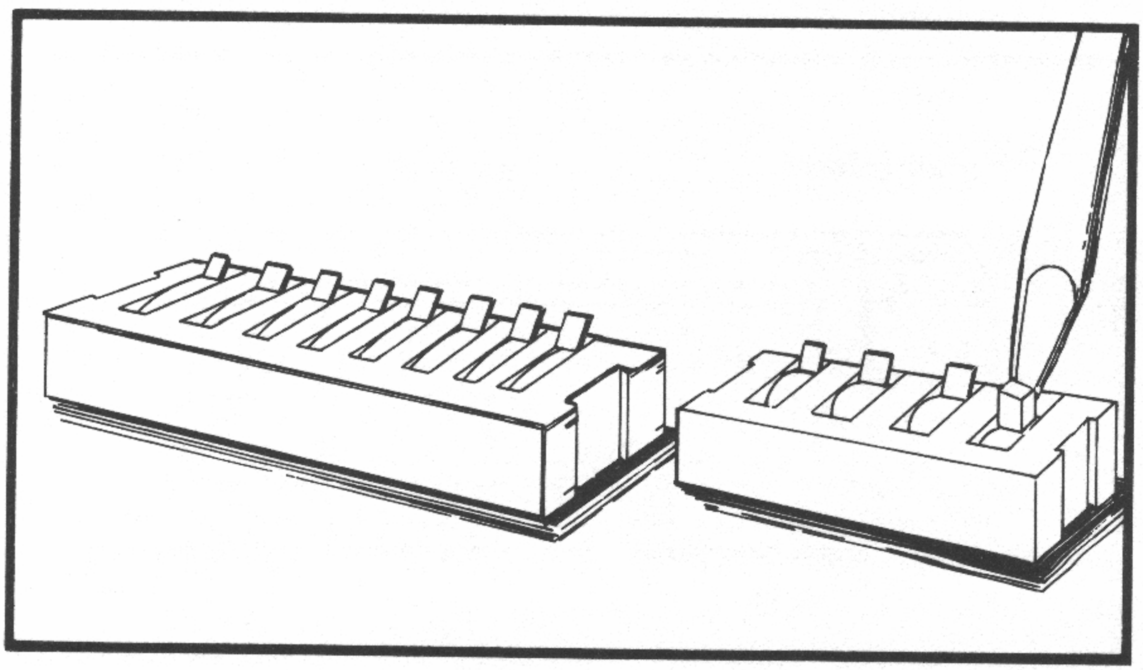


FIGURE 13.2. The correct way to change a DIP switch setting

RX Series

LOCATING THE DIP SWITCHES

The DIP switches in RX Series printers are located inside the case. To locate the DIP switch settings, the upper case must be removed. The following steps should be undertaken to access the DIP switches:

Step 1. Remove all paper from the printer.

Step 2. Turn the power OFF.

Step 3. Unplug the printer.

Step 4. Remove the manual paper feed knob by pulling it straight out.

Step 5. Remove the paper separator.

Step 6. Remove the lid.

Step 7. Remove the two screws that attach the upper case to the rest of the printer. The screws require a Phillips screwdriver. Take care to not misplace the screws. Figure 13.3 depicts the position of the screws. It may be necessary to remove the ribbon cartridge before removing the screws.

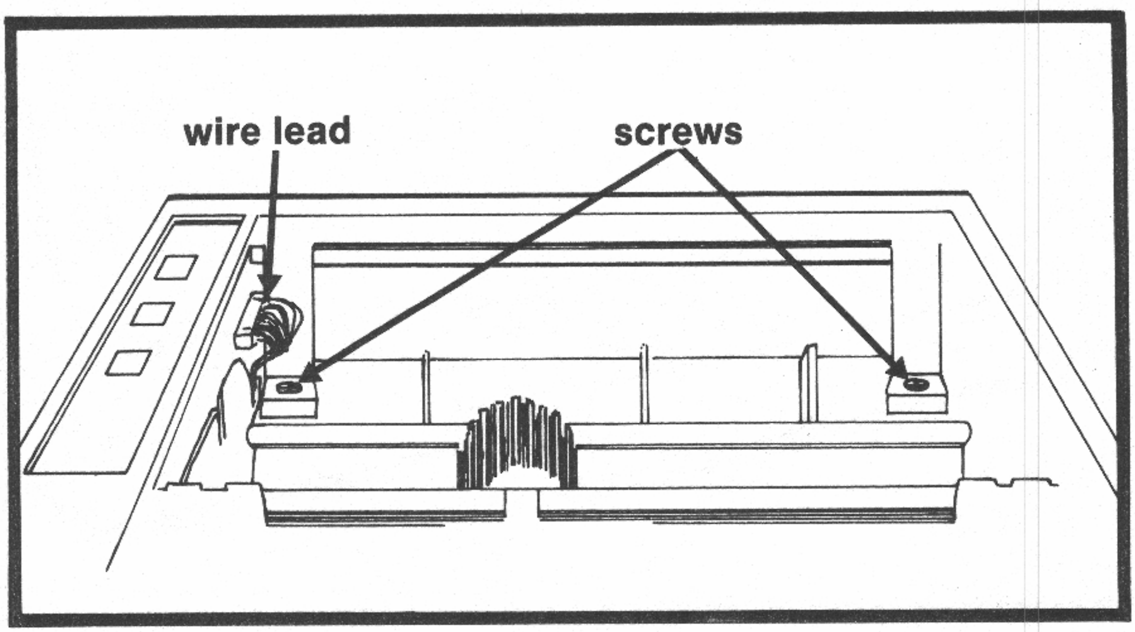


FIGURE 13.3. Location of screws attaching upper case and wire lead to the control panel

Step 8. Detatch the wires leading to the control panel by gently pulling out. The position of these wires is indicated in figure 13.3.

Step 9. Remove the upper case by lifting the front edge and pulling the entire case upward.

Step 10. Find the DIP switches inside the printer. Figure 13.4 indicates the location of the switches.

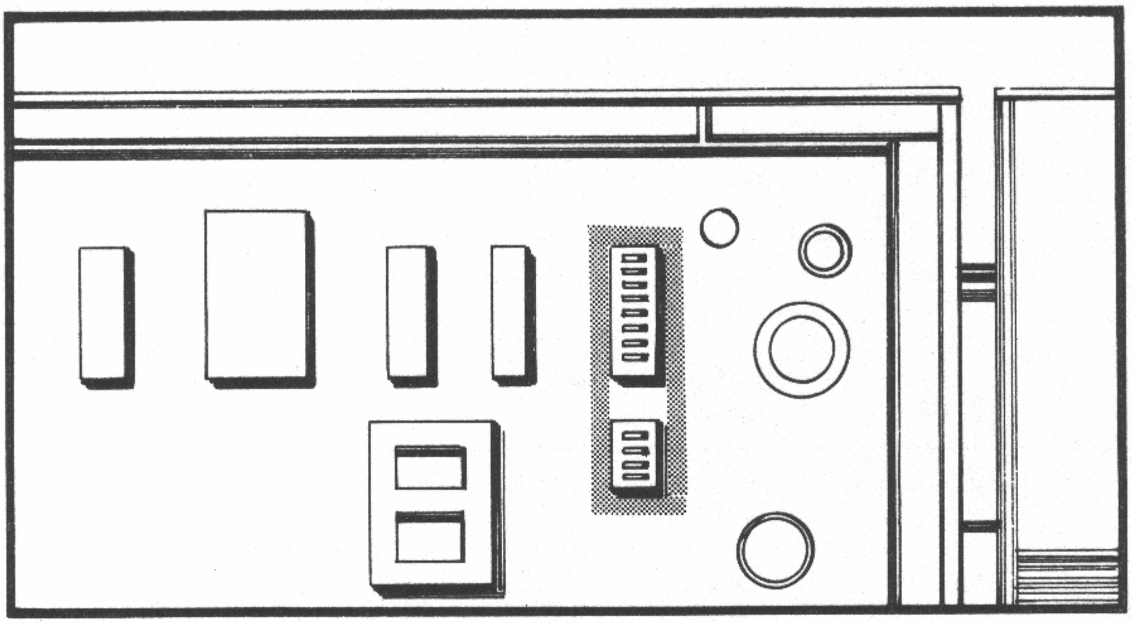


FIGURE 13.4. Locating the DIP switches

After setting the DIP switches, reassemble the printer by performing the removal instructions in reverse order. Care should be taken when plugging the wires back into the control panel. Be sure that the plug is lined up with the receptacle.

SETTING DIP SWITCHES TO SELECT MODES

The RX Series printers have two DIP switches. Switch one has eight pins. Switch two has four pins. These twelve pins control ten printer functions. Tables 13.2 and 13.3 summarize the possible settings for each switch.

Table 13.2 DIP switch one for the RX Series

Pin Number	Factory Set Condition	Function Controlled by Pin	Condition Set by ON	Condition Set by OFF
1-1	Pica-sized characters	Selection of character width	Condensed characters	Pica-sized characters
1-2	Control codes	Selects graphic symbol mode	Graphic symbols	Control codes
1-3	Buzzer will sound	Sounding of the buzzer	Buzzer will not sound	Buzzer will sound
1-4	11 inch form length	Selection of form length	12 inch form length	11 inch form length
1-5	Sensor active	Activation of paper-out sensor	Sensor not active	Sensor active
1-6	USA character set*	Selection of international character set	See table 13.4	
1-7	USA character set	Selection of international character set	See table 13.4	
1-8	USA character set	Selection of international character set	See table 13.4	

* The factory setting will most likely be other than USA if the printer was sold outside of the United States.

Table 13.3. DIP switch two for the RX Series

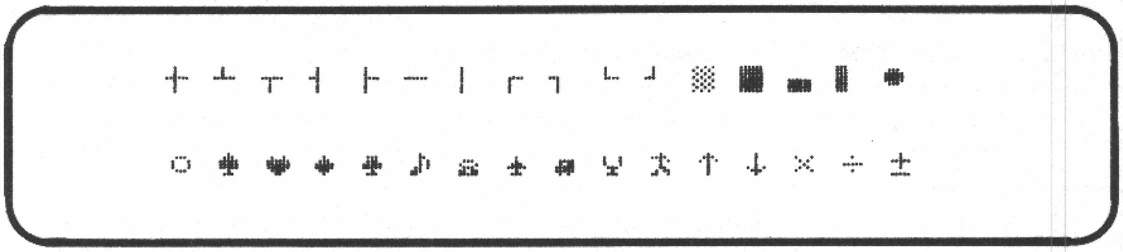
Pin Number	Factory Set Condition	Function Controlled by Pin	Condition Set by ON	Condition Set by OFF
2-1	"0" for zero character	Selection of zero font	"Ø" for zero character	"0" for zero character
2-2	Printer always selected	Specification of printer select	Printer always selected	Host computer must select printer
2-3	No automatic line feed	Selection of automatic line feed	Automatic LF with each CR	No automatic line feed
2-4	Skip-over inactive	Activation of perforation skip-over	Skip-over active	Skip-over inactive

Pin one of switch one controls the print mode. The print mode defines character width. Setting the pin ON causes condensed characters to be selected. There are 17 condensed characters per inch. Setting the pin OFF causes pica-sized characters to be output. There are 10 pica-sized characters per inch.

```
These are pica-sized characters

These are condensed characters
```

Pin two of switch one controls the graphic symbols mode. Setting the pin ON causes graphics symbols corresponding to control codes 128 through 159 on the ASCII table to be selected. Setting the pin OFF causes the standard control codes in 128 through 159 on the ASCII table to be selected.

The graphic symbols set is depicted below:



Pin three of switch one controls the buzzer. Setting the pin ON causes the buzzer to be disabled. Setting the pin OFF allows the buzzer to function. If a mechanical error occurs, the buzzer will sound without regard to the setting of this pin.

Pin four of switch one controls the form length. The form length is the amount of space between consecutive perforations. Setting the pin ON causes a 12 inch form length to be selected. Setting the pin OFF causes an 11 inch form length to be selected.

Pin five of switch one controls activation of the paper-out sensor. When the sensor is active and the end of the paper is detected, the printer will fall off line and light the paper out indicator. After the printer has been refilled with paper, output will resume when the on line button is depressed. Setting the pin ON deactivates the sensor. Setting the pin OFF activates the sensor.

Pins six, seven, and eight of switch one together control selection of an international character set. International character sets allow letters and symbols that are not used in the USA to be printed. Character sets for eight countries can be specified using these pins. Three additional sets are available, but these must be accessed using control codes. Table 13.4 shows how to set the three pins for each country.

Table 13.4 Designating international character sets

Country	Switch one		
	Pin 6	**Pin 7**	**Pin 8**
USA	ON	ON	ON
France	ON	ON	OFF
Germany	ON	OFF	ON
England	ON	OFF	OFF
Denmark I	OFF	ON	ON
Sweden	OFF	ON	OFF
Italy	OFF	OFF	ON
Spain	OFF	OFF	OFF

In the international character sets, a different letter or symbol always replaces a USA symbol. The first row of table 13.5 shows the USA symbols that may be different in an international character set. The other rows show the characters that would replace the USA symbols for each set.

To output an international character using your USA keyboard, find the international character in table 13.5. Then, scan up the column to the USA row. Strike the character which appears in the USA row. The printer will "translate" the character to the international character from the set that is presently specified. For example, while the English character set is selected, sending a "#" results in a "£" being printed.

Table 13.5. International character sets

COUNTRY												
USA	#	$	@	[	\	]	^	`	{	¦	}	~
FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
ENGLAND	£	$	@	[	\	]	^	`	{	¦	}	~
DENMARK	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~

Pin one of switch two controls selection of the zero font. The zero font specifies how the printer will output the character for zero. Setting the pin ON selects "Ø" as the character for a zero. Setting the pin OFF selects "0" as the character for a zero.

Pin two of switch two controls selection of the printer. When the printer is selected and is on line, it is ready to receive data and produce output. When the printer is not selected, the host computer must select it by sending a select message before sending any data. The printer will not receive and process data while it is not selected. Setting the pin ON causes the printer to always be selected. Setting the pin OFF causes the printer to be not selected until it receives a select message from the host computer.

Pin three of switch two controls the automatic line feed function. Line feed can be handled by one of two schemes. The host computer can send both a carriage return and a line feed each time a new line is needed. Or, the host computer can send only a carriage return and allow the printer to automatically add the line feed. Setting the pin ON causes an automatic line feed for each carriage return. Setting the pin OFF causes no automatic line feeds. If the printer always double-spaces, change the pin to OFF. If the printer outputs all the data on one line, change the pin to ON. If the printer operates as it should, leave the pin alone.

Pin four of switch two controls the perforation skip-over function. When the perforation skip-over function is active, the printer will skip the bottom inch of the form and start output at the top of the next form. To have the skip function work properly, be sure that the top-of-form marked in the printer's memory coincides with the physical top-of-form on the paper. One way to accomplish this alignment is to be sure that the first line of a form is under the print head when the power to the printer is turned ON. Setting the pin ON activates the perforation skip-over function. Setting the pin OFF deactivates this feature.

FX Series

LOCATING THE DIP SWITCHES

The DIP switches in the FX Series are simple to access. The switches are under the vent cover which is located in the upper right-hand corner, above the control board. Figure 13.5 indicates the location of the vent cover. The following steps should be undertaken to access the DIP switches:

Step 1. Remove the screw holding the vent cover in place. The position of the screw is indicated in figure 13.5. The screw will require a Phillips screw driver.

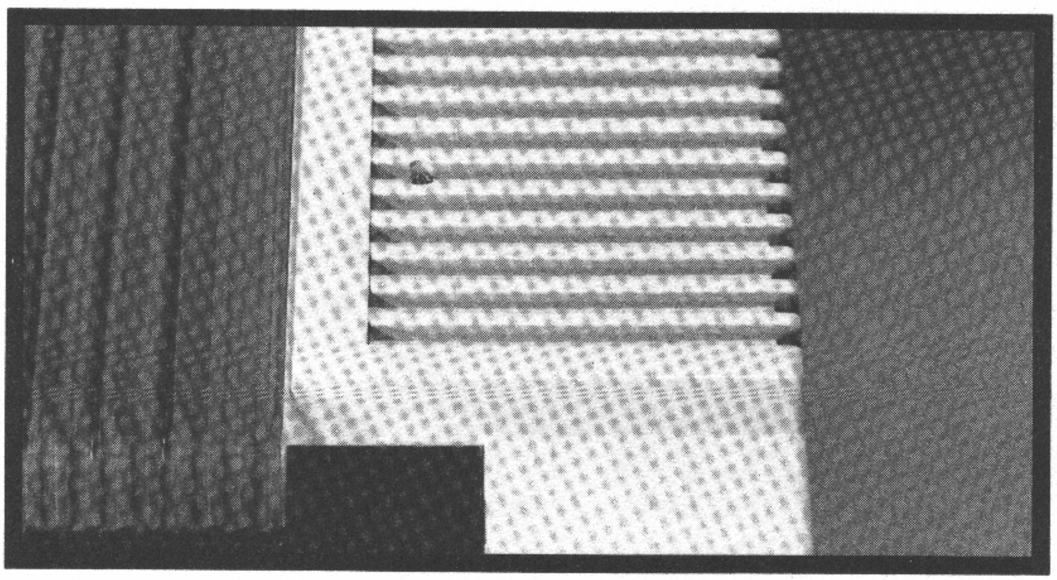


FIGURE 13.5. Location of vent cover and screw holding it in place

Step 2. Remove the vent cover by pushing down on the top and pulling sideways. By inserting a coin in the edge of the vent and twisting, the vent will come off more easily. Figure 13.6 illustrates the procedure.

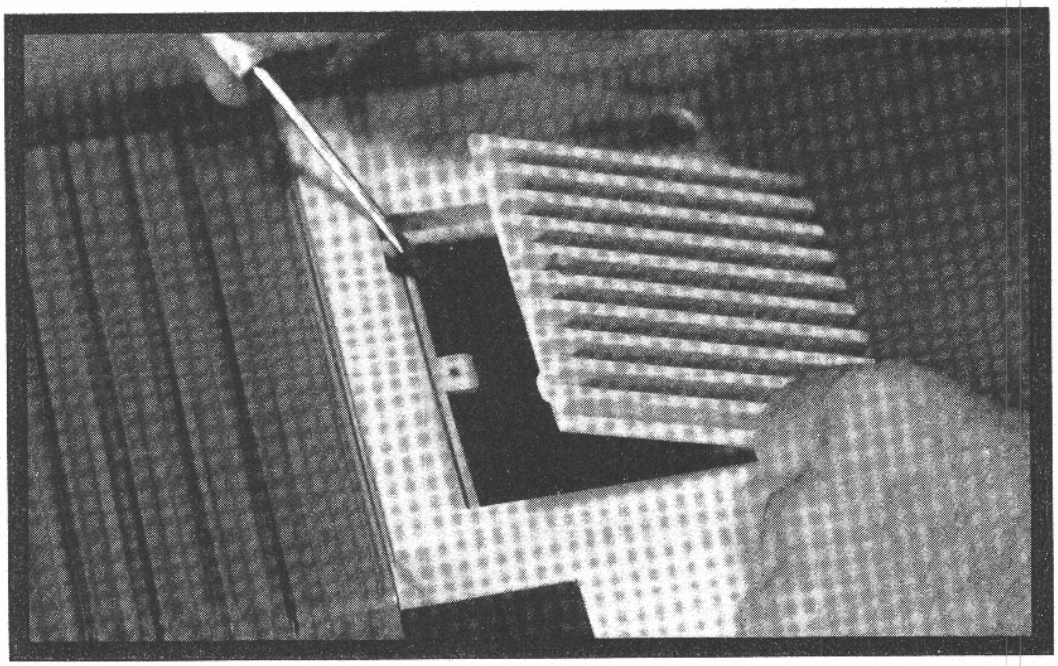


FIGURE 13.6. Removing the vent cover

Step 3. Be sure that the printer is OFF before changing any of the DIP switches.

SETTING DIP SWITCHES TO SELECT MODES

The FX Series printers have two DIP switches. Switch one has eight pins. Switch two has four pins. These twelve pins control ten printer functions. Tables 13.6 and 13.7 summarize the possible settings for each switch.

Table 13.6. DIP switch one for the FX Series

Pin Number	Factory Set Condition	Function Controlled by Pin	Condition Set by ON	Condition Set by OFF
1-1	Pica-sized characters	Compressed mode	Compressed characters	Pica-sized characters
1-2	"0" for zero character	Selection of zero font	"Ø" for zero character	"0" for zero character
1-3	Sensor active	Activation of paper-out sensor	Sensor not active	Sensor active
1-4	User defined characters	Use of RAM memory	2K buffer	User-defined characters
1-5	Normal	Print weight	Emphasized	Normal
1-6	USA character set *	Selection of international character set	See table 13.8	
1-7	USA character set	Selection of international character set	See table 13.8	
1-8	USA character set	Selection of international character set	See table 13.8	

Table 13.7. DIP switch two for the FX Series

Pin Number	Factory Set Condition	Function Controlled by Pin	Condition Set by ON	Condition Set by OFF
2-1	Printer always selected	Specification of printer select	Printer always selected	Host computer must select printer
2-2	Buzzer will sound	Sounding of the buzzer	Buzzer will sound	Buzzer will not sound
2-3	Skip-over inactive	Activation of perforation skip-over	Skip-over active	Skip-over inactive
2-4	No automatic line feed	Selection of automatic line feed	Automatic LF with each CR	No automatic line feed

* The factory setting will most likely be other than USA if the printer was sold outside of the United States.

Pin one of switch one controls the compressed mode. The compressed mode defines character width. Setting the pin ON causes compressed characters to be selected. There are 17.16 compressed characters per inch. Setting the pin OFF causes pica-sized characters to be selected. There are 10 pica-sized characters per inch.

```
These are pica-sized characters

These are condensed characters
```

Pin two of switch one controls selection of the zero font. The zero font specifies how the printer will output the character for a zero. Setting the pin ON selects "Ø" as the character for a zero. Setting the pin OFF selects "0" as the character for a zero.

Pin three of switch one controls activation of the paper-out sensor. When the sensor is active and the end of the paper is detected, the printer will fall off line and light the paper out indicator. After the printer is refilled with paper, output will resume when the on line button is depressed. Setting the pin ON deactivates the sensor. Setting the pin OFF activates the sensor.

Pin four of switch one controls use of the RAM memory. Setting the pin ON causes the RAM memory to be used as a 2K buffer. The larger buffer makes communication between the host computer and the printer quicker and smoother. Setting the pin OFF causes the RAM memory to be employed to store user defined characters. If the user is not defining characters, the pin should be set ON.

Pin five of switch one controls the print weight. Print weight defines the print density of the character. Setting the pin ON causes the emphasized mode to be selected. The emphasized mode is denser than the normal mode. Setting the pin OFF causes the normal mode to be selected.

```
This is the normal weight mode

This is the emphasized weight mode
```

Pins six, seven, and eight of switch one together control selection of an international character set. International character sets allow letters and symbols that are not used in the USA to be printed. Character sets for eight countries can be specified using these pins. One additional set is available, but this must be accessed using control codes. Table 13.8 shows how to set the three pins for each country.

Table 13.8. Designating international character sets

	Switch One		
Country	**Pin 6**	**Pin 7**	**Pin 8**
USA	ON	ON	ON
France	ON	ON	OFF
Germany	ON	OFF	ON
England	ON	OFF	OFF
Denmark	OFF	ON	ON
Sweden	OFF	ON	OFF
Italy	OFF	OFF	ON
Spain	OFF	OFF	OFF

The different letter or symbol always replaces a USA symbol. The first row of table 13.9 shows the USA symbols that may be different in an international character set. The other rows show the character that replaces the USA symbol for each set.

To output an international character using your USA keyboard, find the international character in table 13.9. Then, scan up the column to the USA row. Strike the character which appears in the USA row. The printer will "translate" the character to the international character from the set that is presently specified. For example, while the English character set is selected, sending a "#" results in a "£" being printed.

Table 13.9. International character sets

COUNTRY												
USA	#	$	@	[	\	]	^	`	{	¦	}	~
FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
ENGLAND	£	$	@	[	\	]	^	`	{	¦	}	~
DENMARK	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~

Pin one of switch two controls selection of the printer. When the printer is selected and is on line, it is ready to receive data and produce output. When the printer is not selected, the host computer must select it by sending a select message before sending any data. The printer will not receive and process data while it is not selected. Setting the pin ON causes the printer to always be selected. Setting the pin OFF causes the printer to be not selected until receiving a select message from the host computer.

Pin two of switch two controls the buzzer. Setting the pin ON allows the buzzer to function. Setting the pin OFF causes the buzzer to be disabled.

Pin three of switch two controls the perforation skip-over function. When the perforation skip-over function is active, the printer will skip the bottom inch of the form and start output at the top of the next form. To have the skip function work properly, be sure that the top-of-form marked in the printer's memory coincides with the physical top-of-form on the paper. One way to accomplish this alignment is to be sure the first line of a form is under the print head when the power to the printer is turned ON. Setting the pin ON activates the perforation skip-over function. Setting the pin OFF deactivates this feature.

Pin four of switch two controls the automatic line feed function. Line feed can be handled by one of two methods. The host computer can send both a carriage return and a line feed each time a new line is needed. Or, the host computer can send only a carriage return and allow the printer to automatically add the line feed. Setting the pin ON causes an automatic

line feed for each carriage return. Setting the pin OFF causes no automatic line feeds. If the printer always double-spaces, change the pin to OFF. If the printer outputs all the data on one line, change the pin to ON. If the printer operates as it should, leave the pin alone.

14

Bit Image Graphics

Introduction

In this chapter, we will discuss the modes that control bit image graphics. Bit image graphics refers to the printer's ability to generate a figure by firing individual pins.

In the first four sections, we will discuss general information necessary to carry out bit image graphics. In the remaining sections, we will discuss the individual control codes used to specify the bit image graphics density. The density of bit image graphics is a measure of how many dots are printed in an inch. The output will appear darker as the density increases.

Reserving Space

Bit image graphics are generated one space at a time. In bit image graphics, a space is one dot in width and between 1 and 9 dots in height. The height of an individual space will be discussed in detail in the next section.

Before a bit image graphics mode can be activated, the total number of spaces across the graphics figure must be specified. All of the bit image graphics modes use the same scheme to represent this information to the printer.

The last two parameters in the BASIC expression for a bit image graphics control code define the number of spaces reserved for bit image graphics. These parameters take the following form:

$$\text{CHR\$}(b_1)\text{CHR\$}(b_2)$$

where b_1 and b_2 are numbers or numerical expressions. The value specified for b_1 represents the number of individual spaces. The range of b_1 is from 0 to 255. The value specified for b_2 represents the number of 256 space groups.

The total width specified by b_1 and b_2 is:

$$\text{Total width} = b_1 + (b_2 \times 256)$$

This value can never exceed the maximum number of spaces on a single line. The highest allowable values for b_2 are listed in table 14.1. The value of b_2 can range from 0 to the maximum allowable.

TABLE 14.1. Maximum values of b_2 for different densities

MODEL		DENSITY					
		Single 160 dpi *	Double 120 dpi	Quadruple 240 dpi	One-to-one 72 dpi	CRT Screen 90 dpi	Epson QX-10 80 dpi
80	maximum spaces per line	480	960	1,920	576	720	640
	maximum value of b_2	1	3	7	2	2	2
100	maximum spaces per line	816	1,632	3,264	979	1,224	1,088
	maximum value of b_2	3	6	12	3	4	4

* dots per inch

To determine the specification for a graphic that will be 672 spaces wide, calculate $672 \div 256$. The result is 2 with a remainder of 160. The values specified for b_1 and b_2 should be:

$$b_1 = 160$$
$$b_2 = 2$$

The total width value also specifies how much data the printer must receive in order to print a graphics figure. All data received will be treated as bit image data until the printing of the graphics figure has been completed. We will discuss bit image data in a later section.

Computers generally keep track of the number of characters output on the current output line. The computer will automatically send a carriage return and a line feed when the line is full. The default width of an output line is generally 80 characters. When a bit image mode is active, the computer will count each data item as one character.

Bit image graphics can easily exceed 80 spaces in width. When the carriage return (ASCII code 13) and the line feed (ASCII code 10) are encountered, they will be treated as bit image data. This results in 2 undesired spaces being output in the graphics figure. To avoid this mistake, the width of the computer's output line must be changed to a larger value. The BASIC command to accomplish this alteration has the form:

WIDTH *b*

or

WIDTH *device name, b*

where *b* is a number or numerical expression. The value of *b* defines a new output line width. Defining a value of 255 for *b* will cause the carriage return and line feed to be suppressed. Generally, the *device name* specification for the printer is:

"LPT1:"

Some computers require that the *device name* be specified with the WIDTH command while others do not.

Mixing Text and Graphics

Both text and graphics can be generated on the same line. However, each text character reduces the amount of the space available for graphic output. Table 14.2 lists the number of graphics spaces required by the characters in each of the different pitch modes. For example, each expanded elite sized character will occupy 20 of the double density graphics spaces on the line.

TABLE 14.2. Number of graphics spaces used by one character

Density	PITCH					
	Pica	Expanded Pica	Elite	Expanded Elite	Condensed	Expanded Condensed
Single	6	12	5	10	3.5	7
Double	12	24	10	20	7	14
Quadruple	24	48	20	40	14	28

Firing Pins

The print head has nine individual dot pins. In one space, any combination of these nine pins can be fired. However, special measures must be taken to fire the bottom pin of the print head. We will discuss the scheme for firing nine pins in the last section of this chapter.

The top eight pins are each assigned a numerical value. The numerical value associated with each pin is shown in figure 14.1.

top pin → 8 - 128
7 - 64
6 - 32
5 - 16
4 - 8
3 - 4
2 - 2
1 - 1
bottom pin → 0 - unused

FIGURE 14.1. Value associated with each pin

When a bit image mode is active, the printer will treat all data as bit image data. That is, each data item will be treated as a numerical value. If a character is sent to the printer, the printer uses that character's corresponding ASCII code to define the graphics output. The character itself is not output. The following discussion explains the relationship between bit image data and which pins are fired.

SINGLE PINS

A numerical value equal to a value associated with one of the pins will cause that pin to be fired. For example, if a 32 is received by the printer, pin 6 will be fired. A value of 0 will cause no pins to be fired.

MULTIPLE PINS

A numerical value that is not equal to a value associated with one of the pins will cause more than one pin to be fired. The unique set of pins whose sum of associated values is equal to the data value are fired. For example, a data value of 129 will cause pins 1 and 8 to be fired. The only combination of associated pin values that equals 129 is 128 and 1.

Figure 14.2 contains examples of how to determine the data value to fire any combination of pins. The values associated with the pins to be fired are added. The result is the data value that will cause the desired pins to be fired.

The height of the line can be easily controlled. In the previous examples, the height of the line was 8 dots. A line height of 7 dots is obtained by allowing the data values to range from 0 to 127, rather than 0 to 255. Users of computers that control only 7 of the data lines on their parallel printer interface are limited to firing only the middle 7 pins of the print head.

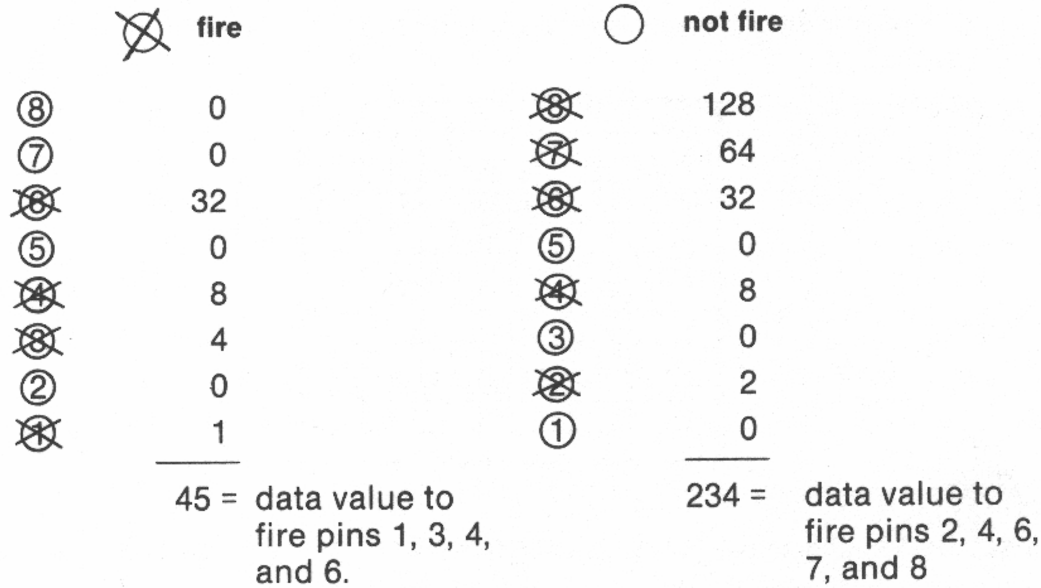
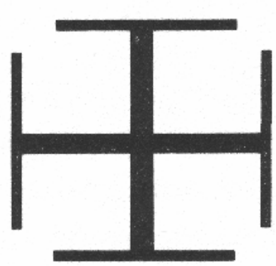


FIGURE 14.2. Examples illustrating how to obtain the data value to fire several pins

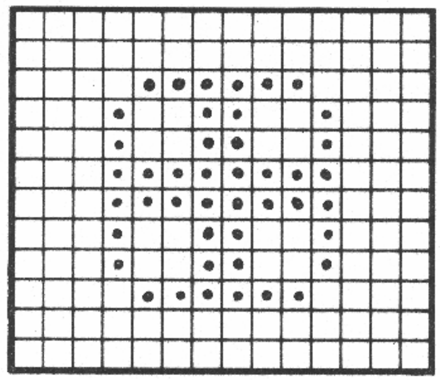
Planning a Simple Graphic

This section includes an example illustrating the process for planning a simple graphic. We will use the graphic planned here in the example programs illustrating each of the individual bit image graphics modes. The example programs in the next chapter will show how to plan and execute more complex graphics.

Suppose that we wanted the printer to generate the following figure:



The first step is to plan the figure on graph paper:



Next, the data value for each space must be calculated. Note that there are only three different configurations of dots:

8		0	8	•	128	8	•	128
7	•	64	7		0	7	•	64
6	•	32	6		0	6	•	32
5	•	16	5	•	16	5	•	16
4	•	8	4	•	8	4	•	8
3	•	4	3		0	3	•	4
2	•	2	2		0	2	•	2
1		0	1	•	1	1	•	1
		126			153			255

The data string to create the desired symbol is: 126, 153, 153, 255, 255, 153, 153, 126. We will use this data string in the examples in this chapter.

Single Density

The single density bit image graphics mode provides 60 dots per inch. All densities are measured with reference to horizontal distance. In the single density mode, none of the dots will overlap. Figure 14.3 shows how the graphics figure planned earlier would appear when generated using single density graphics.

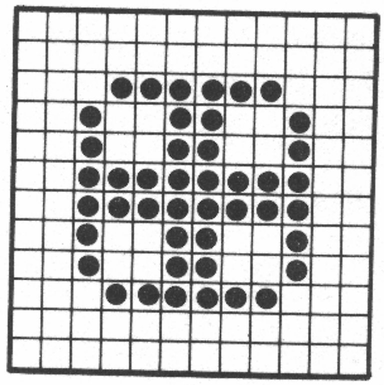


FIGURE 14.3. Several spaces of single density bit image graphics output

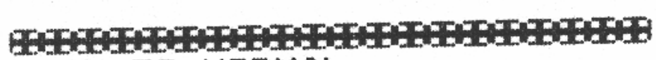
The ESC K code activates the single density graphics mode. The BASIC expression for ESC K is:

CHR$(27)"K"CHR$(b_1)CHR$($b_2$)

where the values of b_1 and b_2 specify the width of the graphics figure as discussed in the section on reserving space. All bit image graphics modes will be deactivated when the specified amount of data has been received. An example program illustrating the use of the single density bit image graphics mode follows:

```
10 WIDTH 255
20 LPRINT CHR$(27)"K"CHR$(160)CHR$(0);
30 FOR I=1 TO 20
40 FOR J=1 TO 8
50 READ A
60 LPRINT CHR$(A);
70 NEXT J
80 RESTORE
90 NEXT I
100 WIDTH 80
110 LPRINT
120 LPRINT "BACK TO NORMAL"
130 END
140 DATA 126,153,153,255
150 DATA 255,153,153,126
```

The program output is:



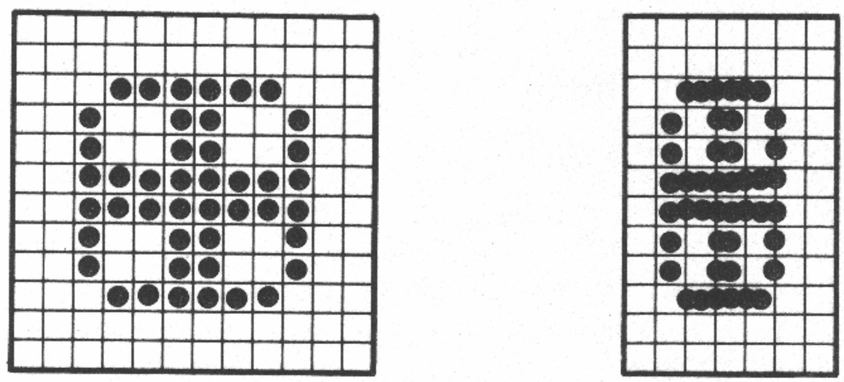
```
BACK TO NORMAL
```

The statement in line 10 caused the automatic carriage return and line feed from the host computer to be suppressed. The command in line 100 returned the output line width to the standard 80 characters.

The statement in line 20 enabled the single density bit image graphics mode for a width of 160 spaces. The nested loops in lines 30 to 90 caused the simple graphic that we planned earlier to be generated 20 times on the same line. Note that each time the graphic was generated, 8 data items were sent to the printer. Repeating the graphic 20 times caused 160 data items to be sent to the printer. The bit image graphics mode was automatically deactivated when the reserved space was filled. The RESTORE command allowed the data list to be reused. This reuse saved us from inputting the same data 20 times.

Double Density

The double density bit image graphics mode provides 120 dots per inch. Figure 14.4 shows how the graphic we planned earlier would appear when generated using double density graphics.



single

double

FIGURE 14.4. Comparison of single and double density bit image graphics output

The ESC L code activates the double density graphics mode. The BASIC expression for ESC L is:

CHR$(27)"L"CHR$(b_1)CHR$($b_2$)

where the values of b_1 and b_2 specify the width of the graphic as discussed in the section on reserving space. An example program illustrating the use of the double density bit image graphics mode follows:

```
10 WIDTH 255
20 LPRINT "SINGLE:";
30 LPRINT CHR$(27)"K"CHR$(160)CHR$(0);
40 GOSUB 110
50 LPRINT "DOUBLE:";
60 LPRINT CHR$(27)"L"CHR$(160)CHR$(0);
70 GOSUB 110
80 WIDTH 80
90 LPRINT "BACK TO NORMAL"
100 END
110 FOR I=1 TO 20
120 FOR J=1 TO 8
130 READ A
140 LPRINT CHR$(A);
150 NEXT J
160 RESTORE
170 NEXT I
180 LPRINT
190 RETURN
200 DATA 126,153,153,255
210 DATA 255,153,153,126
```

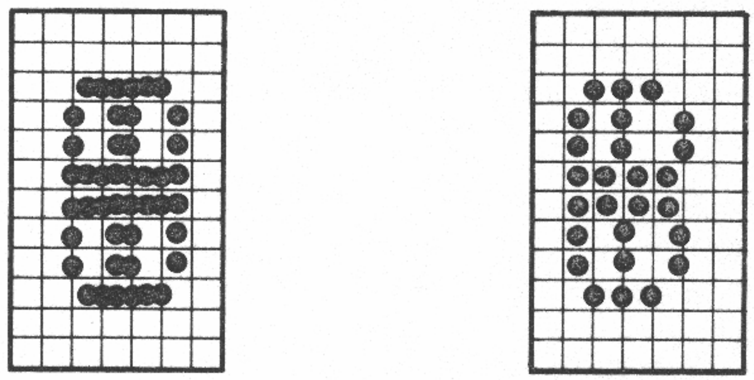
The program output is:

```
SINGLE:
DOUBLE:
BACK TO NORMAL
```

Compare the two densities. Note that the double density output is darker and one-half as wide as the single density output. Also note that text and graphics may be mixed on one line. The statement in line 60 enabled the double density bit image graphics mode.

Double-Speed, Double Density

The double-speed, double density bit image graphics mode provides 120 dots per inch. This is the same density as the double density mode. The difference between the two modes is the speed at which the print head moves while generating the output. The print head moves twice as fast in the double-speed mode. But, the same pin cannot generate a dot in consecutive spaces. Figure 14.5 shows how the graphics figure we planned earlier would appear when generated using double-speed, double density graphics.



normal-speed **double-speed**

FIGURE 14.5. Comparison of the normal and double-speed, double density bit image graphics modes*

The ESC Y code activates the double-speed, double density graphics mode. The BASIC expression for ESC Y is:

CHR\$(27)"Y"CHR\$(b_1)CHR\$(b_2)

* Note that none of the overlapping dots were printed in the double-speed mode.

where the values of b_1 and b_2 specify the width of the graphic as discussed in the section on reserving space. An example program illustrating the use of the double-speed, double density bit image graphics mode follows:

```
10 WIDTH 255
20 LPRINT "SINGLE:";
30 LPRINT CHR$(27)"L"CHR$(160)CHR$(0);
40 GOSUB 110
50 LPRINT "DOUBLE:";
60 LPRINT CHR$(27)"Y"CHR$(160)CHR$(0);
70 GOSUB 110
80 WIDTH 80
90 LPRINT "BACK TO NORMAL"
100 END
110 FOR I=1 TO 20
120 FOR J=1 TO 8
130 READ A
140 LPRINT CHR$(A);
150 NEXT J
160 RESTORE
170 NEXT I
180 LPRINT
190 RETURN
200 DATA 126,153,153,255
210 DATA 255,153,153,126
```

The program output is:

```
SINGLE:
DOUBLE:
BACK TO NORMAL
```

Compare the output generated by the two different double density speed modes. Note that some of the dots are not printed when the double-speed version is active. Graphics that are to be generated using the double-

speed, double density bit image graphics mode should be carefully planned. The planning should take into account the fact that horizontally adjacent dots cannot be printed. The command in line 60 activated the double-speed, double density bit image graphics mode.

Quadruple Density

The quadruple density bit image graphics mode provides 240 dots per inch. In the quadruple density mode, horizontally adjacent dots cannot be printed. Figure 14.6 shows how the graphic we planned earlier would appear when generated using quadruple density graphics.

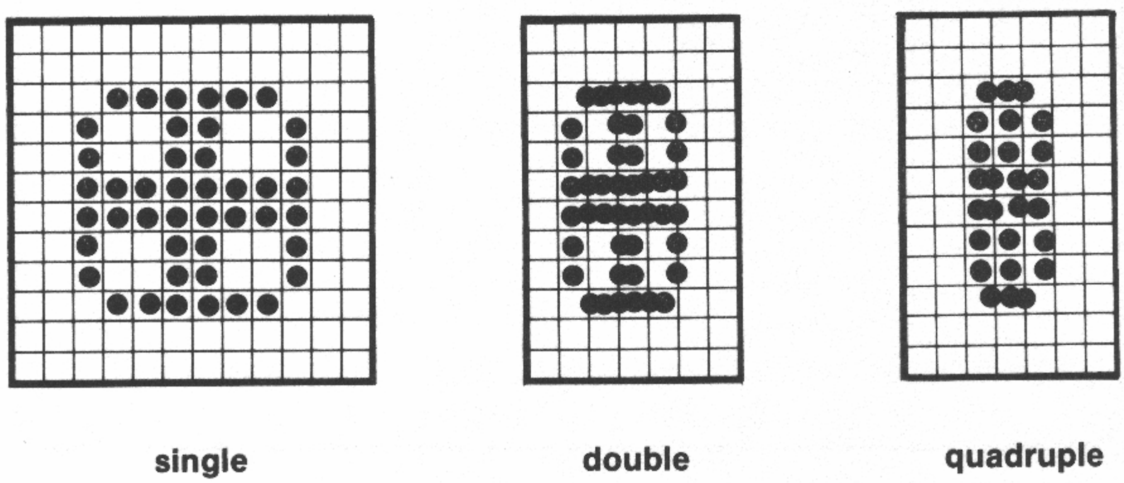


FIGURE 14.6. Comparison of single, double, and quadruple density bit image graphics output

The ESC Z code activates the quadruple density graphics mode. The BASIC expression for ESC Z is:

$$\text{CHR\$(27)“Z”CHR\$}(b_1)\text{CHR\$}(b_2)$$

where the values of b_1 and b_2 specify the width of the graphic as discussed in the section on reserving space. An example program illustrating the use of the quadruple density bit image graphics mode follows:

```
10 WIDTH 255
20 LPRINT "SINGLE:";
30 LPRINT CHR$(27)"K"CHR$(160)CHR$(0);
40 GOSUB 140
50 LPRINT "DOUBLE:";
60 LPRINT CHR$(27)"L"CHR$(160)CHR$(0);
70 GOSUB 140
80 LPRINT "QUADRUPLE:";
90 LPRINT CHR$(27)"Z"CHR$(160)CHR$(0);
100 GOSUB 140
110 WIDTH 80
120 LPRINT "BACK TO NORMAL"
130 END
140 FOR I=1 TO 20
150 FOR J=1 TO 8
160 READ A
170 LPRINT CHR$(A);
180 NEXT J
190 RESTORE
200 NEXT I
210 LPRINT
220 RETURN
230 DATA 126,153,153,255
240 DATA 255,153,153,126
```

The program output is:

```
SINGLE:
DOUBLE:
QUADRUPLE:
BACK TO NORMAL
```

Compare the three densities. Note that not all of the dots were printed in the quadruple density mode. Also, each individual graphic can no longer be easily distinguished in the quadruple density mode. For these reasons, quadruple density graphics should be planned to take into account the

horizontal compression and the fact that horizontally adjacent dots cannot be printed.

Bit Image Select Code

The bit image select code allows selection of any of the available bit image graphics densities. The FX Series has three additional densities besides the four already discussed. The RX Series has two additional densities besides the four already discussed. The two additional densities that the FX and RX Series have in common are both designed for printing screen graphics. The third additional density on the FX Series provides 72 dots per inch. This density allows for a one-to-one correspondence between vertical and horizontal distance.

The ESC * code controls the bit image selection. The BASIC expression for ESC * has the form:

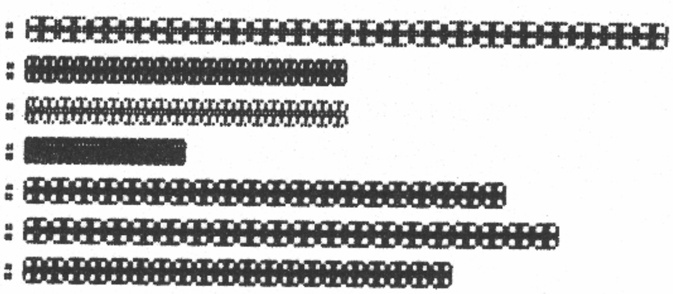
CHR$(27)"*"CHR$(*n*)CHR$($b_1$)CHR$(b_2);

where b_1 and b_2 represent the width of the graphic as discussed in the section on reserving space. And, n is a number or numerical expression from 0 to 6. The value of n specifies which bit image graphics density will be selected. Table 14.3 lists which density will be selected for each n value. An n value of 5 cannot be specified for the RX Series. The following program illustrates the use of the bit image select mode:

```
10 WIDTH 255
20 FOR N=0 TO 6
30 LPRINT N":";
40 LPRINT CHR$(27)"*"CHR$(N)CHR$(160)CHR$(0);
50 GOSUB 100
60 NEXT N
70 WIDTH 80
80 LPRINT "BACK TO NORMAL"
90 END
100 FOR I=1 TO 20
110 FOR J=1 TO 8
120 READ A
130 LPRINT CHR$(A);
140 NEXT J
150 RESTORE
160 NEXT I
170 LPRINT
180 RETURN
190 DATA 126,153,153,255
200 DATA 255,153,153,126
```

The program output is:

```
0 :
1 :
2 :
3 :
4 :
5 :
6 :
BACK TO NORMAL
```



Compare the output generated by the seven different bit image graphics modes. The command in line 40 used the bit image select code to activate the seven modes.

TABLE 14.3. Values of n for the bit image select code

n	Density
0	Single -- (60 dpi)
1	Double -- (120 dpi)
2	Double-Speed, Double Density -- (120 dpi)
3	Quadruple -- (240 dpi)
4	Epson QX-10 Screen -- (80 dpi)
5	One-to-One -- (72 dpi)
6	CRT Screen -- (90 dpi)

Firing the Ninth Pin (Newer RX and FX Series Only)

To fire the ninth pin in the print head while generating bit image graphics, the printer requires two data items for each space in the graphic. So, using all nine pins doubles the data requirements in comparison to using eight pins. The printer uses the first data item to determine which of the top eight pins should be fired just as in any other bit image graphics mode. The printer uses the second data item to decide if the bottom pin should be fired. If the value of the second data item is less than 128, the bottom pin will not be fired. If the value of the second data item is 128 or greater, the bottom pin will be fired.

The ESC ^ code activates the 9 pin bit image mode. The BASIC expression for ESC ^ has the form:

CHR\$(27)"^"CHR\$(n)CHR\$(b_1)CHR\$(b_2);

where b_1 and b_2 represent the width of the graphic as discussed in the section on reserving space. And, n is a number or numerical expression

whose value can be either 0 or 1. If the value of *n* is 0, the 9 pin single density bit image graphics mode is enabled. If the value of *n* is 1, the 9 pin double density bit image graphics mode is activated. The following program illustrates how to use the 9 pin bit image graphics mode:

```
10 WIDTH 255
20 FOR N=0 TO 1
30 LPRINT N":";
40 LPRINT CHR$(27)"^"CHR$(N)CHR$(160)CHR$(0);
50 GOSUB 100
60 NEXT N
70 WIDTH 80
80 LPRINT "BACK TO NORMAL"
90 END
100 FOR I=1 TO 20
110 FOR J=1 TO 16
120 READ A
130 LPRINT CHR$(A);
140 NEXT J
150 RESTORE
160 NEXT I
170 LPRINT
180 RETURN
190 DATA 126,0,153,0,153,0,255,128
200 DATA 255,128,153,0,153,0,126,0
```

The program output is:

```
0 :
1 :
BACK TO NORMAL
```

Note that we modified the graphic slightly to show the firing of the ninth pin. The command in line 40 selected the 9 pin bit image graphic mode. Notice that extra items had to be added to the data list. These extra data items were used to control the firing of the ninth pin.

15

Examples Using Bit Image Graphics

Introduction

In this chapter we will present several example programs utilizing bit image graphics. These programs illustrate the general techniques used to execute more complicated graphics. Developing a program to generate bit image graphics can be tedious and time consuming. However, the payoff is professional looking output.

Generating a Line Graph from Data Points

In this section we show how to use bit image graphics to generate a line graph from several data points. Specifically, we will generate a graph plotting distance against time.

We will use the single density bit image graphics mode for this example. The example program is the same for both the RX and FX Series.

The first step in planning the graph is to set the dimensions. We want the graph to fit on a page of this book. So, the maximum horizontal width is 4-5/8 inches. We will make the graph approximately square. The single density mode provides 60 dots per inch. So, there will be a total of about 276 bit image graphics spaces across the graph. Some of these spaces will be used to generate the margin labels and headings.

We drew a crude sketch to help plan the formatting of the graph. Figure 15.1 contains a copy of the sketch when we were finished planning the format of the graph.

We used the information gained from the sketch to set the line spacing and the limits on the control loops. The example program follows:

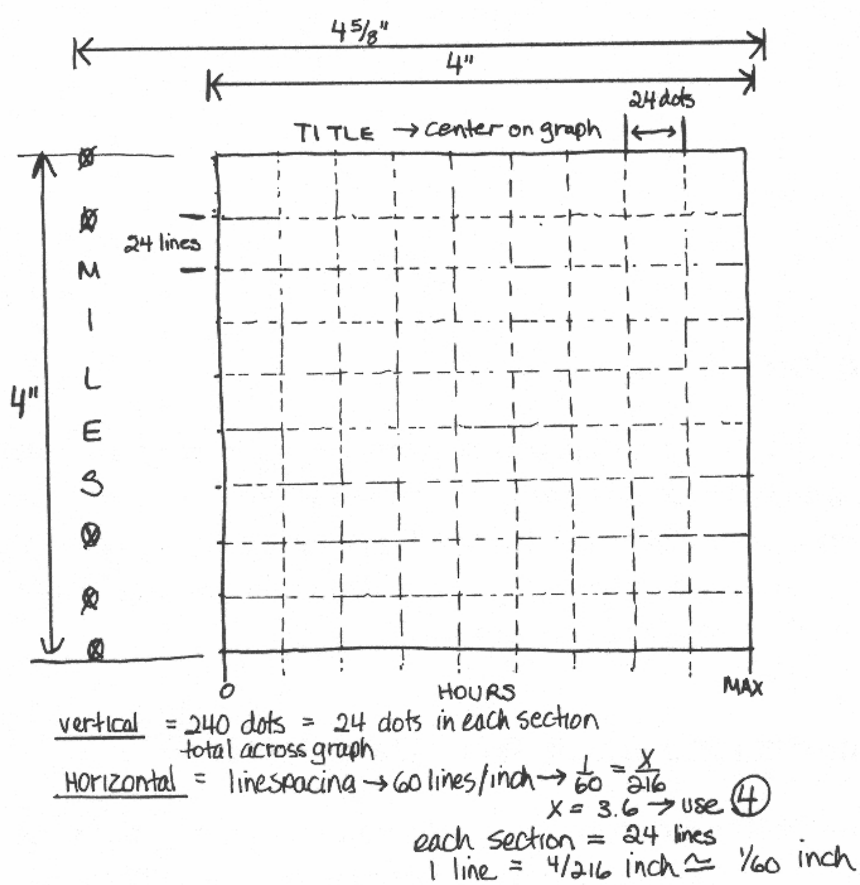


FIGURE 15.1. Copy of the sketch we drew to help plan the format of the graph

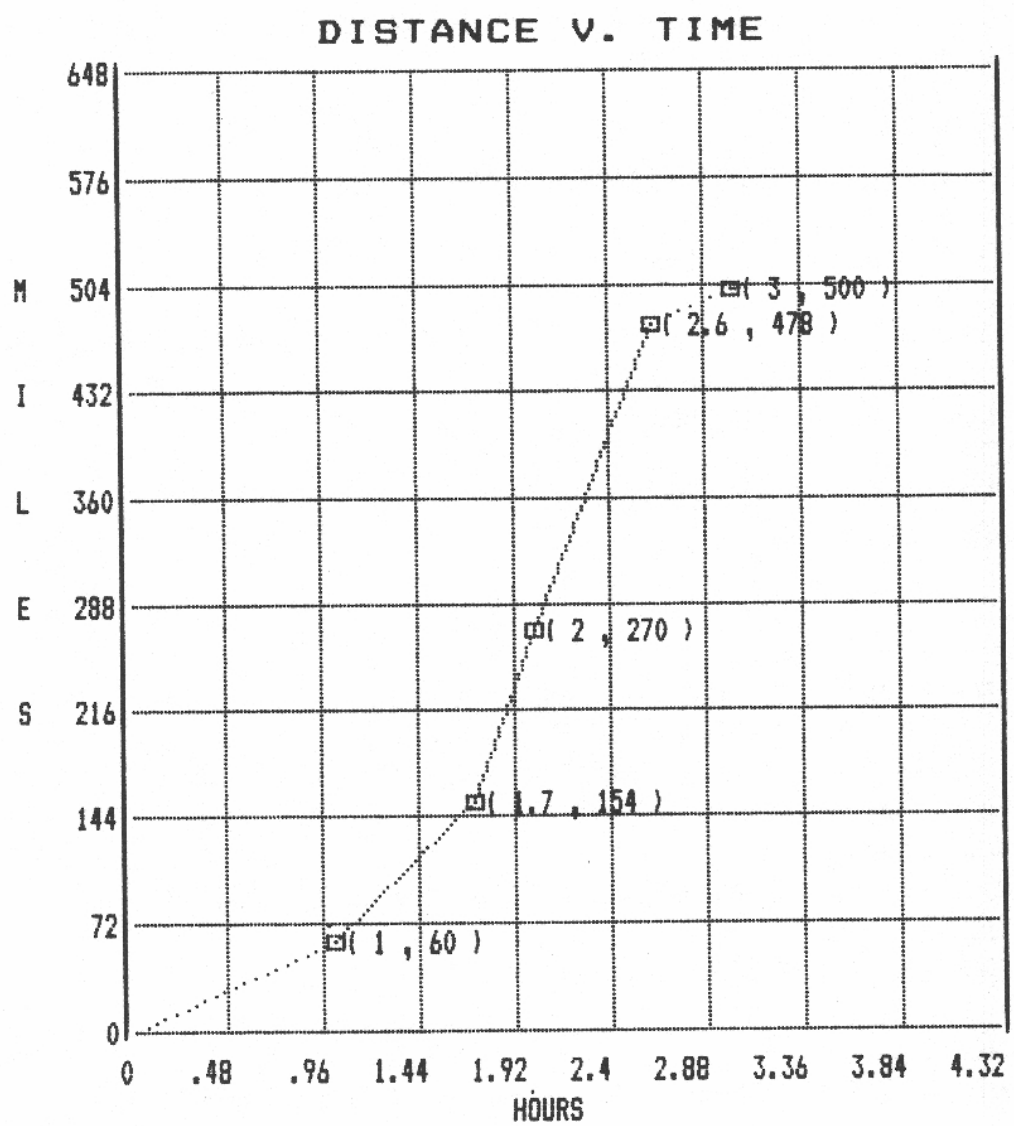
```
10 WIDTH 255
20 READ N
30 DIM DIST(N+1),TIME(N+1)
40 DIST(N+1)=0
50 TIME(N+1)=0
60 FOR K=1 TO N
70 READ DIST(K),TIME(K)
80 NEXT K
90 NOW=0
100 A=0
110 B=0
120 NXT=1
130 INCR=DIST(1)
140 GOSUB 500
150 INCRV=INCR
160 AMNTV=AMNT
170 INCR=TIME(1)
180 GOSUB 560
190 AMNTH=AMNT
200 INCRH=0
210 LPRINT CHR$(15)CHR$(14);
220 LPRINT "                  DISTANCE V. TIME"
230 LPRINT CHR$(27)"3"CHR$(4);
240 GOSUB 410
250 FOR I=1 TO 9
260 FOR J=1 TO 24
270 GOSUB 600
280 GOSUB 880
290 LPRINT
300 INCRV=INCRV-AMNTV
310 NEXT J
320 GOSUB 410
330 NEXT I
340 LPRINT CHR$(27)"2"
350 GOSUB 730
360 LPRINT CHR$(27)"@"
370 WIDTH 80
380 END
390 DATA 5,500,3,478,2.6,270,2,154,1.7,60,1
400 DATA " "," ",M,I,L,E,S," "," "," "
```

```
410 READ MARG$
420 LPRINT MARG$"   ";
430 LPRINT INCRV;
440 LPRINT CHR$(27)"K"CHR$(216)CHR$(0);
450 FOR L=1 TO 216
460 LPRINT CHR$(16);
470 NEXT L
480 LPRINT CHR$(13);
490 RETURN

500 AMNT=INCR/216
510 A%=(AMNT/1)
520 AMNT=A%
530 AMNT=AMNT+1
540 INCR=AMNT*216
550 RETURN

560 AMNT=INCR*100/216
570 A%=AMNT/1
580 AMNT=A%/100+.01
590 RETURN

600 LPRINT "         |";
610 LPRINT CHR$(27)"K"CHR$(216)CHR$(0);
620 FOR O=1 TO 8
630 FOR P=1 TO 23
640 LPRINT CHR$(0);
650 NEXT P
660 LPRINT CHR$(16);
670 NEXT O
680 FOR M=0 TO 24
690 LPRINT CHR$(0);
700 NEXT M
710 LPRINT "|";
720 RETURN

730 LPRINT "         ";
740 INCRH=0
750 LPRINT INCRH;
760 INCRH=INCRH+AMNTH*24
770 FOR Q=1 TO 9
780 IF INCRH<1 THEN LPRINT "   ";
              ELSE LPRINT "  ";
```

```
790 LPRINT INCRH;
800 INCRH=INCRH+AMNTH*24
810 NEXT Q
820 LPRINT
830 FOR Q=1 TO 35
840 LPRINT " ";
850 NEXT Q
860 LPRINT "HOURS"
870 RETURN

880 LPRINT CHR$(13);
890 LPRINT "        ";
900 IF INCRV<(DIST(NXT)+AMNTV/2)
        THEN GOSUB 1050
910 IF A=0 AND B=0 THEN RETURN
920 FIRE=A+B*INCRV
930 INCRH=0
940 Z=0
950 WHILE FIRE>(INCRH+AMNTH/2)
960 Z=Z+1
970 INCRH=INCRH+AMNTH
980 WEND
990 LPRINT CHR$(27)"K"CHR$(Z+1)CHR$(0);
1000 FOR Y=1 TO Z
1010 LPRINT CHR$(0);
1020 NEXT Y
1030 LPRINT CHR$(16);
1040 RETURN

1050 NXT=NXT+1
1060 NOW=NOW+1
1070 B=(TIME(NXT)-TIME(NOW))/
        (DIST(NXT)-DIST(NOW))
1080 A=TIME(NOW)-B*DIST(NOW)
1090 INCRH=0
1100 Z=0
1110 WHILE TIME(NOW)>(INCRH+2.5*AMNTH)
1120 Z=Z+1
1130 INCRH=INCRH+AMNTH
1140 WEND
1150 Z=Z-9
1160 LPRINT CHR$(27)"K"CHR$(Z+5)CHR$(0);
```

```
1170 FOR Y=1 TO Z
1180 LPRINT CHR$(0);
1190 NEXT Y
1200 LPRINT CHR$(124)CHR$(68)CHR$(84)
            CHR$(68)CHR$(124);
1210 LPRINT "("TIME(NOW)","DIST(NOW)")";
1220 LPRINT CHR$(13);
1230 LPRINT "         ";
1240 RETURN
```

The program output is:



The statements in lines 10 through 200 initialized the variables and counters and selected the print mode. The value of *N* specified the number of data points on the graph. The arrays, DIST and TIME, were used to store the values of the data points used to generate the graph. The values must be arranged in descending order. The statements in lines 40 and 50 specified an extra data point at the origin. Inclusion of this point in the data set forced the graph to always start at the point (0,0). The values of *A* and *B* are used to represent the constants in the equation for a straight line:

$$\text{HOURS} = A + B \times \text{MILES}$$

The variables INCRV and INCRH were used as counters. Their values represented the current vertical and horizontal position on the graph, respectively. The variables AMNTV and AMNTH were used to store the value of the vertical and horizontal increment, respectively. Note that we used the compressed pitch mode throughout the program.

The command in line 220 caused the title of the graph to be printed. The statement in line 230 selected the line spacing. We used a line spacing of 4/216 of an inch. The sketch shows how we arrived at this selection. The subroutine called in line 240 caused the top line of the graph to be generated. The nested loops in lines 250 to 330 caused the rest of the graph to be generated. The inner loop generated one line of the graph each time it was executed. The outer loop caused the horizontal scales to be generated.

The statements in lines 340 to 380 caused the labels at the foot of the graph to be generated and returned all modes to their default values.

The subroutine starting at line 410 was used to print the labels at the left of the graph and the horizontal lines across the graph. Note that the lines were generated using the single density bit image graphics mode. The command in line 440 enabled the graphics mode for a width of 216 spaces. The loop in lines 450 to 470 sent 216 data values of 16 to the printer. Recall that a value of 16 will cause the fifth pin to be fired. Note the statement in line 480. This statement caused a carriage return. Returning to the beginning of the line allows another subroutine to add more output to the same line by overstriking whatever was previously printed.

The subroutine starting at line 500 was used to calculate the initial value of INCRV and the value of AMNTV. The subroutine starting in line 560 was used to calculate the value of AMNTH.

The subroutine starting in line 600 was used to print the vertical lines on the graph. The statement in line 610 enabled the graphics mode for 216 spaces. The nested loops in lines 620 to 670 sent 192 data items to the printer. The inner loop sent 23 values of 0 for each of the 8 values of 16 sent by the outer loop. The loop in lines 680 to 700 sent the last 24 values of 0 to the printer.

The subroutine starting at line 730 was used to generate the labels at the foot of the table.

The subroutine starting at line 880 was used to generate the lines between points on the graph. Note that the position on the graph of the dot to form part of the line was calculated by the WHILE loop in lines 950 to 980. The print head was then moved to the correct position by the loop in lines 1000 to 1020. The statement in line 1030 caused the dot to be printed.

Finally, the subroutine starting at line 1050 was used to calculate the values for *A* and *B*. The subroutine also caused each data point, enclosed in a box, to be printed.

Outputting a Figure

Bit image graphics can be used to create a great variety of figures. In this example, we will use an Epson printer to generate a discount coupon.

The first step was to plan the graphic's appearance. Once again we drew a sketch to plan the placement of objects in the figure. Figure 15.2 shows how the sketch looked when we were done planning the coupon.

We decided to use quadruple density graphics. Note that the planning of the graphics took into account the fact that horizontally adjacent dots cannot be output. A width of 4 inches will accommodate 960 quadruple density bit image graphics spaces. We used a maximum line height of 7 dots. The program to generate the coupon follows on page 268:

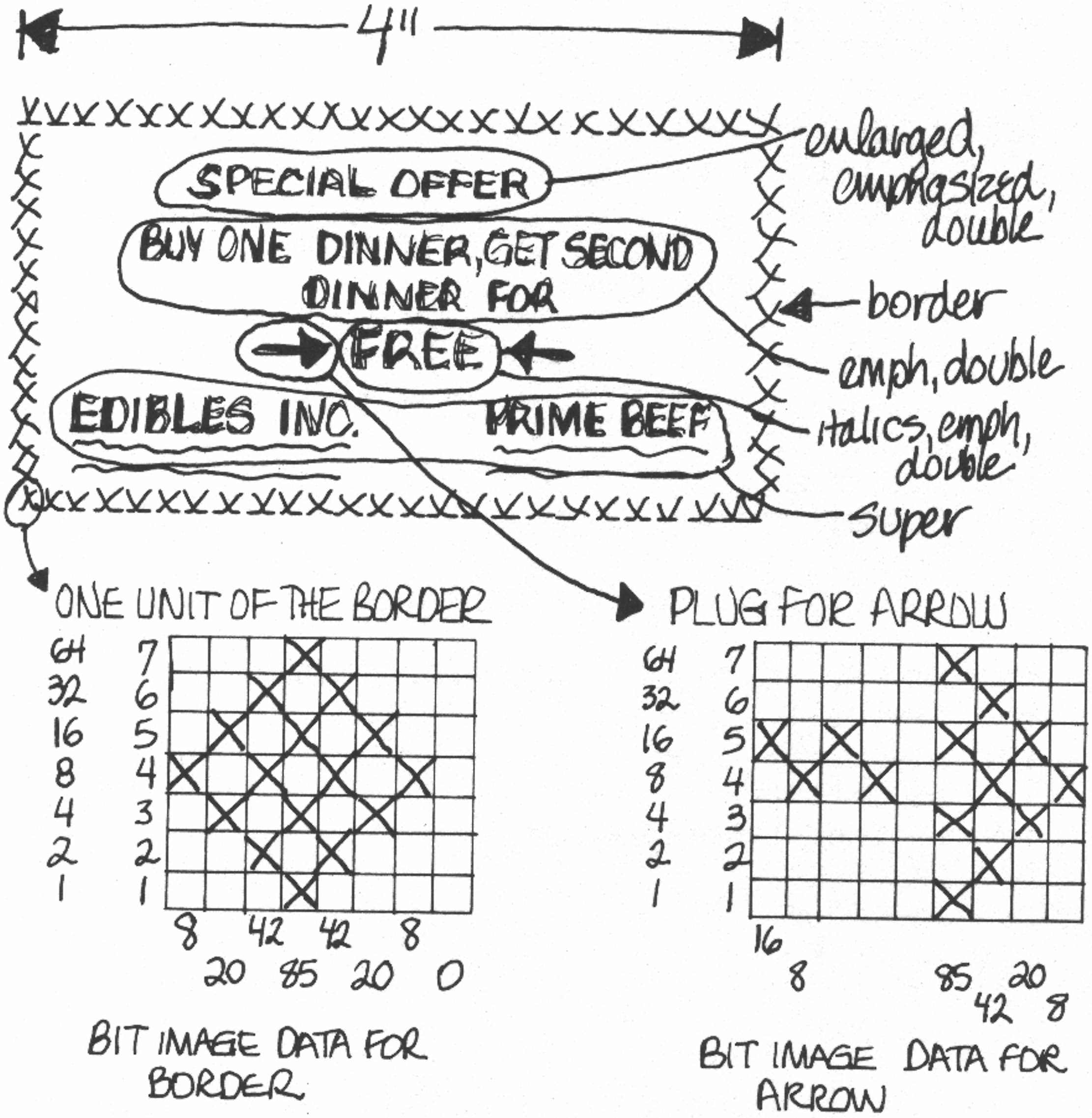


FIGURE 15.2. Sketch used to plan a figure

```
10 WIDTH 255
20 LPRINT CHR$(27)"1";
30 A=1
40 GOSUB 150
50 FOR J=1 TO 14
60 GOSUB 250
70 ON A GOSUB 380,400,460,480,500,
          540,560,590,610,810,
    830,890,940
80 LPRINT
90 NEXT J
100 GOSUB 150
110 WIDTH 80
120 LPRINT CHR$(27)"@"
130 END
140 DATA 8,20,42,85,42,20,8,0

150 LPRINT CHR$(27)"Z"CHR$(192)CHR$(3);
160 FOR K=1 TO 120
170 FOR L=1 TO 8
180 READ BORD
190 LPRINT CHR$(BORD);
200 NEXT L
210 RESTORE
220 NEXT K
230 LPRINT
240 RETURN

250 LPRINT CHR$(27)"Z"CHR$(192)CHR$(3);
260 FOR K=1 TO 2
270 FOR L=1 TO 7
280 READ BORD
290 LPRINT CHR$(BORD);
300 NEXT L
310 RESTORE
320 FOR M=1 TO 946
330 LPRINT CHR$(0);
340 NEXT M
350 NEXT K
360 LPRINT CHR$(13);
370 RETURN
```

```
380 A=A+1
390 RETURN

400 LPRINT CHR$(14);
410 LPRINT CHR$(27)"E";
420 LPRINT CHR$(27)"G";
430 LPRINT "   SPECIAL OFFER";
440 A=A+1
450 RETURN

460 A=A+1
470 RETURN

480 A=A+1
490 RETURN

500 A=A+1
510 LPRINT "       BUY ONE DINNER, ";
520 LPRINT "GET SECOND";
530 RETURN

540 A=A+1
550 RETURN

560 A=A+1
570 LPRINT "              DINNER FOR";
580 RETURN

590 A=A+1
600 RETURN

610 LPRINT CHR$(27)"Z"CHR$(119)CHR$(1);
620 FOR P=1 TO 317
630 LPRINT CHR$(0);
640 NEXT P
650 FOR Q=1 TO 27
660 LPRINT CHR$(16)CHR$(8);
670 NEXT Q
680 LPRINT CHR$(85)CHR$(42)CHR$(20)CHR$(8);
690 LPRINT CHR$(27)"4";
700 LPRINT " FREE ";
```

```
710 LPRINT CHR$(27)"5";
720 LPRINT CHR$(27)"Z"CHR$(58)CHR$(0);
730 LPRINT CHR$(8)CHR$(20)CHR$(42)CHR$(85);
740 FOR R=1 TO 27
750 LPRINT CHR$(8)CHR$(16);
760 NEXT R
770 A=A+1
780 LPRINT CHR$(27)"F";
790 LPRINT CHR$(27)"H";
800 RETURN

810 A=A+1
820 RETURN

830 A=A+1
840 LPRINT CHR$(27)"S1";
850 LPRINT "   EDIBLES,INC.";
860 LPRINT "            ";
870 LPRINT "PRIME BEEF";
880 RETURN

890 A=A+1
900 LPRINT "   24 OVERLOOK DRIVE";
910 LPRINT "      ";
920 LPRINT "FINE SPIRITS";
930 RETURN

940 A=A+1
950 LPRINT "   BAY VILLAGE,OHIO";
960 LPRINT "       ";
970 LPRINT "ON THE LAKE";
980 RETURN
```

The program output is:

```
              SPECIAL  OFFER

       BUY ONE DINNER, GET SECOND
               DINNER FOR
              ⟶ FREE ⟵

  EDIBLES, INC.                  PRIME BEEF
  24 OVERLOOK DRIVE              FINE SPIRITS
  BAY VILLAGE, OHIO              ON THE LAKE
```

The command in line 20 selected 7/72 inch line spacing. We used this spacing so that no white space would appear between lines in the figure. Note that this action caused all the diamonds in the border to appear to touch.

The statements in lines 40 and 100 caused the top and bottom borders to be printed. The loop in lines 50 to 90 caused the balance of the coupon to be generated. Note that each line of the coupon is printed twice by this loop. The subroutine called in line 60 caused the left and right borders to be printed. The subroutines called in line 70 caused the text information appearing on the coupon to be generated.

The subroutine starting in line 150 caused a single unit of the border to be repeated 120 times on the same line. These 120 diamonds were used for the top and bottom borders.

The subroutine starting in line 250 was used to generate the right and left borders. Note the command in line 360. This carriage return prepares the printer to output any text information that will appear on the current line.

The subroutine starting in line 380 was used to skip outputting text on the current line. The subroutines starting in lines 460, 480, 540, 590, and 810 performed an identical action.

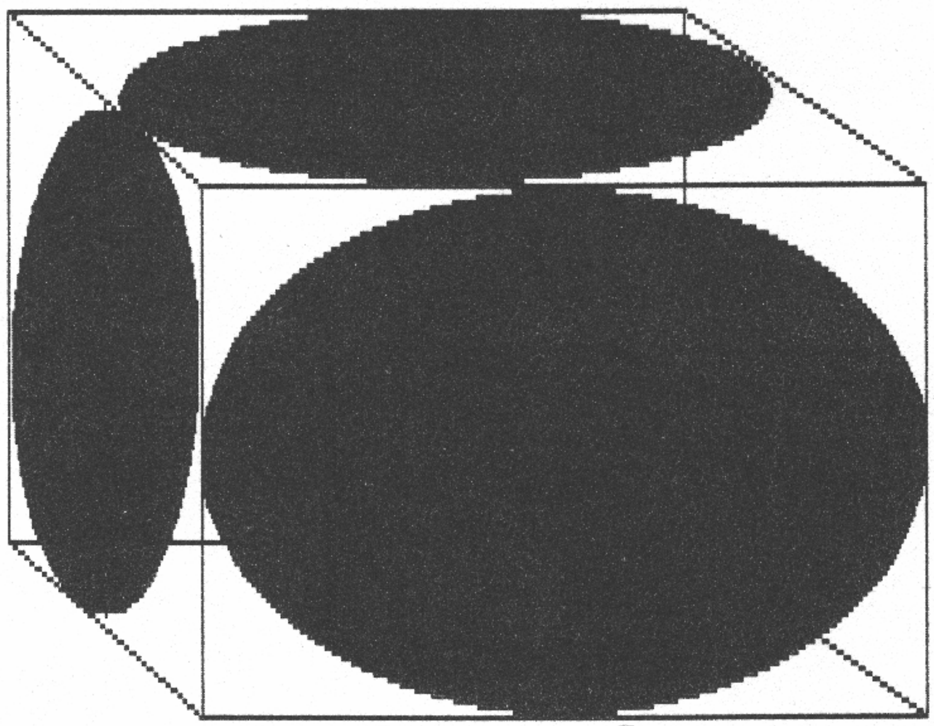
The subroutines starting in lines 400, 500, 560, 610, 830, 890, and 940 were used to output the text of the coupon. Note that the spacing was accomplished by inserting blanks into the LPRINT commands. Each line of text on the coupon was generated using the print styles indicated in figure 15.2.

Printing the Screen

There are occasions when a hard copy of the data that appears on the screen is desired. If your computer system does not provide a routine to dump the screen display to the printer, use the subroutine in the example program to accomplish the task. An example program illustrating a screen dump follows:

```
10 CLS
20 WINDOW (0,0)-(639,199)
30 VIEW (0,0)-(639,199)
40 CIRCLE (350,133),150,,,.33
50 PAINT (350,133)
60 CIRCLE (160,116),48,,,1.27
70 PAINT (160,116)
80 CIRCLE (301,66),135,,,.12
90 PAINT (301,66)
100 LINE (120,50)-(200,83)
110 LINE (120,150)-(200,183)
120 LINE (500,183)-(400,150)
130 LINE (400,50)-(500,83)
140 LINE (500,183)-(200,83),,B
150 LINE (120,150)-(400,50),,B
160 GOSUB 1000
170 END
1000 WIDTH 255
1010 LPRINT CHR$(27)"A"CHR$(2);
1020 FOR I=0 TO 199
1025 LPRINT CHR$(27)"*"CHR$(4)CHR$(128)CHR$(2);
1030 FOR J=0 TO 639
1050 IF COLOR(J,I)=PRESET THEN LPRINT CHR$(0);
                       ELSE LPRINT CHR$(192);
1090 NEXT J
1100 NEXT I
1110 WIDTH 80
1120 LPRINT CHR$(0)
1130 LPRINT CHR$(27)"2";
1140 RETURN
```

The program output is:



Note that the output has been reduced to 80% of the original size. The statements in lines 10 to 150 create a graphic on the screen. The subroutine called in line 160 dumped the graphic to the printer.

The statement in line 1010 selected a line spacing equivalent to two dots. The I FOR...NEXT loop caused each of the 200 rows of pixels on the screen to be sent to the printer. Note that each row of pixels is output as two rows of dots. The J FOR...NEXT loop caused each of the 640 columns of pixels on the screen to be sent to the printer for each row. Note the IF statement in line 1050. The color of an individual pixel was compared to the background color. If that pixel was the same color as the background, a blank space was output. If that pixel was a different color than the background, two dots were output in that space. The command in line 1025 selected the bit image graphics mode. The mode for normal CRT graphics was enabled. A width of 640 was specified.

16

User-Defined Characters (FX Series Only)

Introduction

Only the FX Series allows the user to define his own characters. In this chapter we will discuss all of the information necessary to define and use a unique set of characters.

ROM and RAM Character Sets

ROM stands for Read-Only Memory. The ASCII character set is stored in the printer's ROM. The ROM character set cannot be altered. There are 256 characters in the permanent ROM set.

RAM stands for Random access Memory. The printer's RAM can be used to store a set of user-defined characters or to extend the print buffer. Be sure that DIP switch 1-4 is set OFF before any attempt is made to define characters. Chapter 13 contains information pertaining to the DIP switches. The RAM character set has space for 256 characters. Initially, all of the characters are spaces except for the RAM positions 0 through 31 and 127 through 159. These positions contain the same control characters as positions 0 through 31 and 127 through 159 on the ASCII table.

The RAM character set can be altered. We will discuss how to alter the RAM character set in subsequent sections. The RAM character set is not permanent. Once characters have been defined in RAM, that area can be cleared to spaces by either sending the Master Reset Code or by powering off the printer.

Loading the RAM with the ROM Character Set

The ROM character set can be loaded into the printer's RAM set. This action allows the bulk of the ASCII character set to be used while redefining only a few characters. We will cover the procedure for redefining characters later.

The ESC : code controls loading of the ROM character set into the printer's RAM set. The BASIC expression for ESC : is:

CHR$(27)":"CHR$(b_1)CHR$($b_2$)CHR$(b_3);

where b_1, b_2, and b_3 are numbers or numerical expressions. Presently, the b values have no meaning and should be set to 0. Epson included these parameters for future expansion. An example program using the ESC : code is presented in the next section.

Selecting the RAM Character Set

Generally, the printer uses the ROM character set to generate text output. The RAM character set must be enabled before it can be used to generate text output.

The ESC % code controls selection of the character set. The BASIC expression for ESC % is:

CHR$(27)"%"CHR$(b_1)CHR$($b_2$);

where b_1 and b_2 are numbers or numerical expressions. The value of b_1 specifies which character set to use. If b_1 is 0, the ROM set is selected. If b_1's value is 1, the RAM set is specified. The value of b_2 specifies the area to enable. Presently, a b_2 value of 0 is the only valid area. A b_2 value of 0 specifies an area covering the entire character set. An example program using the ESC : and ESC % codes follows:

```
10 LPRINT "ROM SET"
20 LPRINT CHR$(27)"%"CHR$(1)CHR$(0);
30 LPRINT "RAM SET"
40 LPRINT CHR$(27)":"CHR$(0)CHR$(0)CHR$(0);
50 LPRINT "RAM SET"
60 LPRINT CHR$(27)"%"CHR$(0)CHR$(0);
70 LPRINT "BACK TO ROM SET"
80 END
```

The program output is:

```
ROM SET

RAM SET
BACK TO ROM SET
```

The statement in line 20 selected the RAM character set. Note that the data in the LPRINT command in line 30 did not appear. Instead, a blank line was output. This action occurred because all of the characters of the RAM set were still blank spaces.

The statement in line 40 caused the ROM character set to be loaded into the RAM set. The data in the LPRINT command in line 50 can now be viewed. The command in line 70 enabled the ROM character set.

Planning Characters

Each new character must be planned. The new characters must adhere to the rules governing the formation of characters by Epson printers. These rules are listed below:

1. The character can be a maximum of 8 dots in height.
2. The character can be a maximum of 11 dots in width.
3. No two dots of the character can be located in adjacent positions on the horizontal.
4. Each character requires 12 data items in its specification string. The specification string will be detailed later in this section.

Each character is generated in a 9 x 11 array. The array is horizontally compressed. Figure 16.1 illustrates this compression. If horizontally adjacent dots could be output, they would overlap.

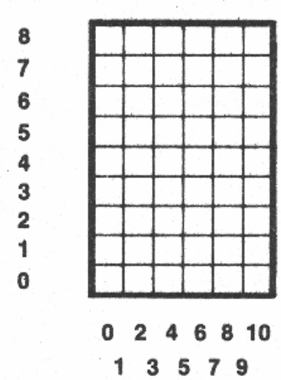


FIGURE 16.1. The 9 x 11 array used to generate a character

The specification string is a list of data items. The data items define a character. The last eleven data items in the specification string are numbers that tell the printer which pins to fire to form the character. The value of these data items is determined in a fashion identical to that used to determine the value of bit image data items. We discussed this material in the section on firing pins in chapter 14.

The first data item in the specification string is called the attribute byte. The attribute byte is used to determine the placement of the character in the 9 x 11 array and the proportional width of the character.

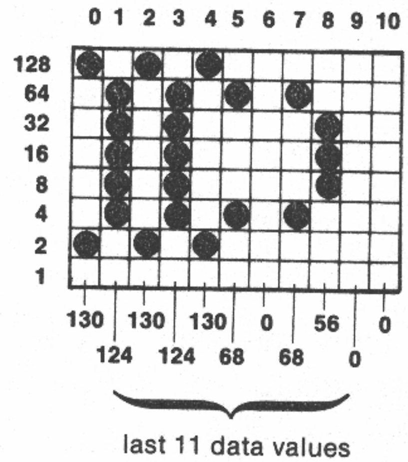
The value of the attribute byte can be determined by summing three numbers. The first number specifies which 8 of the 9 vertical pins should be used to generate the character. If the number is 128, the top eight pins will be used. If the number is 0, the bottom eight pins will be used. The second number specifies the first column of the 9 x 11 array that contains a dot. Table 16.1 lists the values corresponding to different starting columns. The third number specifies the last column in the 9 x 11 array that contains a dot. Table 16.2 lists the values corresponding to different ending columns. If the character is to be printed proportionally, the last column in the 9 x 11 array cannot be the last column in the character. If the character is never going to be printed proportionally, the second and third numbers can be set to 0. Figure 16.2 shows examples of how to determine the specification string for two characters.

Table 16.1. Values corresponding to the different starting columns

Starting Column	Value
0	0
1	16
2	32
3	48
4	64
5	80
6	96
7	112

Table 16.2. Values corresponding to different ending columns

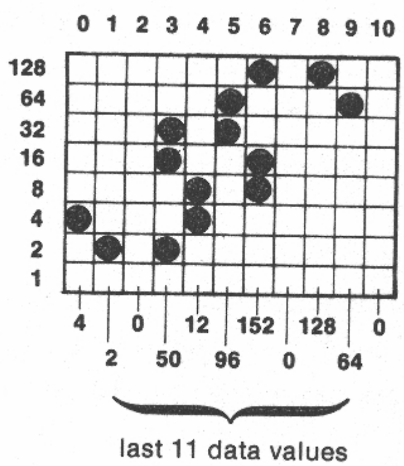
Ending Column	Value
10	——
9	11
8	10
7	9
6	8
5	7
4	6
3	5
2	4
1	3
0	2



Attribute byte:
want printed using top 8 pins: 128
characters starts in column 0: 0
character ends in column 8: 10

138
first data value

Specification string: 138, 130, 124, 130, 124, 130, 68, 0, 68, 56, 0, 0



Attribute byte:
want printed using top 8 pins: 128
characters starts in column 0: 0
character ends in column 9: 11

139
first data value

Specification string: 139, 4, 2, 0, 50, 12, 96, 152, 0, 128, 64, 0

FIGURE 16.2. Examples of how to determine the specification string

Defining Characters in the RAM Set

With the information about specification strings in hand, we are ready to define new characters in the RAM set. For each new character that is defined, the printer requires 12 data items. If we are defining 3 new characters, the printer will treat the next 36 data items as 3 specification strings.

The CHR$ & code controls the defining of new characters. The BASIC expression for CHR$ & can be expressed in two forms:

CHR$(27)"&"CHR$(b)CHR$($c_1$)CHR$(c_2);

or

CHR$(27)"&"CHR$(b)"C_1C_2"

where b, c_1, and c_2 are numbers or numerical expressions. The value of b specifies in which RAM area to store the new character. Stock printers have only one RAM area. This area is specified by a b value of 0.

The values of c_1 and c_2 specify the positions in RAM corresponding to the first and last characters that are to be redefined. The command will cause all characters in the RAM positions between these two values to be redefined. If the two c values are equal, only one character will be redefined. For example, if the value of c_1 was 65 and the value of c_2 was 67, the characters in RAM positions 65 to 67 would be redefined.

C_1 and C_2 in the second form of the ESC & code represent characters. The printer uses the ASCII code corresponding to each character to specify the RAM positions that are to be changed.

For example, if the value of C_1 was "A" and the value of C_2 was "C", the characters in RAM positions 65 to 67 would be redefined. Recall that an "A" corresponds to an ASCII code 65 and a "C" corresponds to an ASCII code 67. Note that the control characters cannot be modified in this fashion.

An example program illustrating the use of the ESC & code follows:

```
10 LPRINT CHR$(27)"%"CHR$(1)CHR$(0);
20 LPRINT CHR$(27)"&"CHR$(0)CHR$(65)CHR$(66);
30 FOR I= 1 TO 24
40 READ N
50 LPRINT CHR$(N);
60 NEXT I
70 LPRINT CHR$(65)CHR$(66)
80 LPRINT CHR$(27)"%"CHR$(0)CHR$(0)
90 END
100 DATA 138,130,124,130,124,130,68,
         0,68,56,0,0
110 DATA 139,4,2,0,50,12,96,152,0,
        128,64,0
```

The program output is:

```
DS
```

These are the two characters we planned in figure 16.2. The statement in line 10 enabled the RAM character set. The statement in line 20 called for 2 new characters to be defined in RAM positions 65 and 66. The loop in lines 30 to 60 sent the two specification strings to the printer. These data strings appear in the DATA statements in lines 100 and 110. The command in line 70 caused the 2 new characters to be printed. The statement in line 80 selected the ROM character set.

Using the New Characters

The new characters that we defined could be used as replacements for the normal "D" and "S" or as special characters. Each of the following two example programs illustrates one of these possible usages of the new characters.

AS REPLACEMENTS

To use the new characters as replacements, the ROM character set must be loaded into the RAM area. The "D" and the "S" will then be redefined using the new characters. The RAM character set is then used to generate all output. A program illustrating this technique follows:

```
10 LPRINT CHR$(27)":"CHR$(0)CHR$(0)CHR$(0);
20 LPRINT CHR$(27)"&"CHR$(0)"DD";
30 GOSUB 1000
40 LPRINT CHR$(27)"&"CHR$(0)"SS";
50 GOSUB 1000
60 LPRINT CHR$(27)"%"CHR$(1)CHR$(0);
70 LPRINT "DAVE SULLIVAN"
80 LPRINT "DS"
90 LPRINT "NOTE THAT ALL OCCURRENCES OF A"
100 LPRINT "'D' OR 'S' ARE GENERATED USING"
110 LPRINT "THE NEW CHARACTERS"
120 LPRINT CHR$(27)"@";
130 LPRINT "BACK TO NORMAL--DD,SS"
140 END
150 DATA 138,130,124,130,124,130,68,
         0,68,56,0,0
160 DATA 139,4,2,0,50,12,96,152,0,
         128,64,0
1000 FOR I=1 TO 12
1010 READ N
1020 LPRINT CHR$(N);
1030 NEXT I
1040 RETURN
```

The program output is:

```
DAVE SULLIVAN
DS
NOTE THAT ALL OCCURRENCES OF A
'D' OR 'S' ARE GENERATED USING
THE NEW CHARACTERS
BACK TO NORMAL--DD,SS
```

The command in line 10 caused the ROM character set to be loaded into the RAM area. The statements in lines 20 and 40 caused the "D" and the "S" to be redefined, respectively. The subroutine called in lines 30 and 50 sent the 12 data items in a specification string to the printer. The command in line 60 selected the RAM character set.

Note that the normal "D" or "S" was always sent to the printer by the LPRINT commands in lines 70 to 110. The output was generated using the new characters because the "D" and the "S" had been redefined.

The statement in line 120 sent the Master Reset Code to the printer. This action caused the RAM character set to be erased and enabled the ROM character set.

AS SPECIAL CHARACTERS

To use the new characters as special characters, the new characters must be defined in the RAM area. The ROM character set will be used to generate most of the output. The RAM character set will be used only when one of the special characters is desired. A program illustrating this technique follows:

```
10 LPRINT CHR$(27)"&"CHR$(0)"12";
20 FOR I=1 TO 24
30 READ N
40 LPRINT CHR$(N);
50 NEXT I
60 SPCD$=CHR$(27)+"%"+CHR$(1)+CHR$(0)+"1"+
         CHR$(27)+"%"+CHR$(0)+CHR$(0)
70 SPCS$=CHR$(27)+"%"+CHR$(1)+CHR$(0)+"2"+
         CHR$(27)+"%"+CHR$(0)+CHR$(0)
80 LPRINT SPCD$"AVID "SPCS$"ULLIVAN"
90 LPRINT SPCD$SPCS$
100 LPRINT "NOTE THAT ONLY THE OCCURRENCES"
110 LPRINT "THAT ARE SPECIFIED AS SPECIAL"
120 LPRINT "OUTPUT A 'D' OR 'S' AS '"SPCD$"'"
130 LPRINT "OR '"SPCS$"'"
140 END
150 DATA 138,130,124,130,124,130,68,
         0,68,56,0,0
160 DATA 139,4,2,0,50,12,96,152,0,
         128,64,0
```

The program output is:

```
DAVID SULLIVAN
DS
NOTE THAT ONLY THE OCCURRENCES
THAT ARE SPECIFIED AS SPECIAL
OUTPUT A 'D' OR 'S' AS 'D'
OR 'S'
```

The statement in line 10 caused the characters in RAM positions 49 and 50 to be defined as the 2 new characters. Note that a "1" corresponds to an ASCII code of 49 and a "2" corresponds to an ASCII code of 50. The loop in lines 20 to 50 sent the data items in the 2 specification strings.

The statement in line 60 defined the character string SPCD$. This definition contains the code to enable the RAM character set, a "1" that corresponds to the new "D" character, and the code to select the ROM character. The statement in line 70 defines a similar string for the new "S" character. Note how simple it is to output one of the special characters using one of these strings. The LPRINT command in line 90 caused both of the special characters to be output.

The ROM character set was used to generate all of the output except for the special characters. The RAM set was enabled only when one of the special characters was to be output.

Appendix A.
ASCII Character Set for Epson Printers

Note that control characters are printed using compressed type. Other characters are printed using pica type.

DECIMAL CODE	CHARACTER
0	
1	
2	
3	
4	
5	
6	
7	BEL
8	BS
9	HT
10	LF
11	VT
12	FF
13	CR
14	SO
15	SI
16	
17	DC1
18	DC2
19	DC3
20	DC4
21	
22	
23	
24	
25	
26	
27	ESC
28	

DECIMAL CODE	CHARACTER
29	
30	
31	
32	SPACE
33	!
34	"
35	#
36	$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9

DECIMAL CODE	CHARACTER
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z

DECIMAL CODE	CHARACTER
91	[
92	\
93	]
94	^
95	_
96	`
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{

DECIMAL CODE	CHARACTER
124	¦
125	}
126	~
127	DEL
128	
129	
130	
131	
132	
133	
134	
135	BEL
136	BS
137	HT
138	LF
139	VT
140	FF
141	CR
142	SO
143	SI
144	
145	DC1
146	DC2
147	DC3
148	DC4
149	
150	
151	
152	
153	
154	
155	ESC
156	

DECIMAL CODE	CHARACTER
157	
158	
159	
160	SPACE
161	!
162	"
163	#
164	$
165	%
166	&
167	'
168	(
169	)
170	*
171	+
172	,
173	-
174	.
175	/
176	0
177	1
178	2
179	3
180	4
181	5
182	6
183	7
184	8
185	9
186	:
187	;
188	<
189	=

DECIMAL CODE	CHARACTER
190	>
191	?
192	@
193	A
194	B
195	C
196	D
197	E
198	F
199	G
200	H
201	I
202	J
203	K
204	L
205	M
206	N
207	O
208	P
209	Q
210	R
211	S
212	T
213	U
214	V
215	W
216	X
217	Y
218	Z
219	[
220	\
221	]
222	^

DECIMAL CODE	CHARACTER
223	_
224	`
225	a
226	b
227	c
228	d
229	e
230	f
231	g
232	h
233	i
234	j
235	k
236	l
237	m
238	n
239	o
240	p
241	q
242	r
243	s
244	t
245	u
246	v
247	w
248	x
249	y
250	z
251	{
252	|
253	}
254	~
255	

Appendix B.
Control Codes for Epson Printers

The control codes for Epson printers are listed in the following tables. The codes are grouped according to their function. Note that if your computer cannot send lowercase letters, use the corresponding ASCII code from appendix A.

Print Pitch

Feature	Control Code	Page
Compressed mode on	CHR$(15)	105
Compressed mode off	CHR$(18)	105
Elite mode on	Escape M	104
Elite mode off	Escape P	104
Expanded mode (continuous) on	Escape W1	108
Expanded mode off	Escape W0	108
Expanded mode (one line) on	CHR$(14)	107
Expanded mode (one line) off	CHR$(20)	107
Pica mode	default	104

Print Weight

Feature	Control Code	Page
Double-strike mode on	Escape G	113
Double-strike mode off	Escape H	113
Emphasized mode on	Escape E	115
Emphasized mode off	Escape F	115
Proportional mode on	Escape p1	119
Proportional mode off	Escape p0	119

Master Codes

Feature	Control Code	Page
Master Select	Escape ! + specification	128
Master Reset	Escape @	101

Special Print Modes

Feature	Control Code	Page
Italics on	Escape 4	139
Italics off	Escape 5	139
Subscript on	Escape S1	136
Subscript off	Escape T	136
Superscript on	Escape S0	135
Superscript off	Escape T	135
Underline on	Escape –1	134
Underline off	Escape –0	134

Miscellaneous Printer Features

Feature	Control Code	Page
Backspace	CHR$(8)	140
Sound Bell	CHR$(7)	147
Cancel text in buffer	CHR$(24)	101
Delete last text character	CHR$(127)	102
Graphics characters	Escape m+spec	142
Half speed mode on	Escape s1	144
Half speed mode off	Escape s0	144
High order bit on	Escape >	149
High order bit off	Escape =	149
High order bit normal	Escape #	149
Immediate mode on	Escape i1	146
Immediate mode off	Escape i0	146
Select international character set	Escape R + spec.	143
Enable printer to receive data (default)	CHR$(17)	---
Disables printer from receiving data	CHR$(19)	---

Line Feeds

Feature	Control Code	Page
Line feed	CHR$(10)	162
Carriage Return	CHR$(13)	161
Set 1/6 inch line spacing (default)	Escape 2	155
Set 1/8 inch line spacing	Escape 0	156
Set 7/72 inch line spacing	Escape 1	157
Set *b*/72 inch line spacing	Escape A CHR$(*b*)	158
Set *b*/216 inch line spacing	Escape 3 CHR$(*b*)	160
One time line feed	Escape J CHR$(*b*)	163
One time reverse line feed	Escape j CHR$(*b*)	164

Forms Control

Feature	Control Code	Page
Form feed	CHR$(12)	169
Set form length in inches	Escape C CHR$(0) CHR$(*b*)	170
Set form length in lines	Escape C CHR$(*b*)	170
Paper out sensor on	Escape 8	175
Paper out sensor off	Escape 9	175
Skip over perforation on	Escape N CHR$(*b*)	173
Skip over perforation off	Escape O	173

Formatting

Feature	Control Code	Page
Horizontal tab	CHR$(9)	184, 186
Set horizontal tab stops	Escape D + specs.	186
or	Escape e0 CHR$ (*b*)	184
Vertical tab	CHR$(11)	179, 181
Set vertical tab stops	Escape B + specs.	181
or	Escape e1 CHR$ (*b*)	179
Set vertical tab stops in channels	Escape b + specs.	182
Select a channel	Escape /	183
Set left margin	Escape l CHR$(*b*)	189
Set right margin	Escape Q CHR$(*b*)	188
Unidirectional mode on	Escape U1	191
Unidirectional mode off	Escape U0	191
One line unidirectional mode on	Escape <	191

Bit Image Graphics

Feature	Control Code	Page
Single density	Escape K + specifications	247
Double density	Escape L + specifications	249
Double density, doublespeed	Escape Y + specifications	251
Quadruple density	Escape Z + specifications	253
Select density code	Escape * + specifications	255
9 pin, single density	Escape ^ CHR$(0) + specs	257
9 pin, double density	Escape ^ CHR$(1) + specs	257

User-Defined Characters

Feature	Control Code	Page
Load ROM set into RAM	Escape : CHR$(0) CHR$(0) CHR$(0)	276
Enable RAM set	Escape % CHR$(1) CHR$(0)	277
Enable ROM set	Escape % CHR$(0) CHR$(0)	277
Define a new character	Escape & + specifications	282

Index

U

W

ABOUT WEBER SYSTEMS, INC.

In 1982, Weber Systems Inc., a software and systems consulting firm based in Cleveland, Ohio, started a publishing division specializing in books about personal computers.

Weber Systems initially published three books. Within a year, their list included eighteen titles. Over twenty more titles are scheduled for publication this year.

The books in the USER'S HANDBOOK series provide clearly presented and immediately useful information about microcomputer hardware and software products. They are created by an editorial staff with extensive experience in computer science, technical writing, and the practical uses of personal computers. The basic tenets of the Weber Systems publishing philosophy are quality, timeliness, and maintenance.

Other Books in This Series
Published by Ballantine Books:

APPLE MACINTOSH USER'S HANDBOOK

APPLE IIe USER'S HANDBOOK

COLECO ADAM USER'S HANDBOOK

COMMODORE 64 USER'S HANDBOOK

COMPAQ USER'S HANDBOOK

IBM BASIC USER'S HANDBOOK

IBM PC & XT USER'S HANDBOOK

IBM PCjr USER'S HANDBOOK

KAYPRO USER'S HANDBOOK

LOTUS 1-2-3 USER'S HANDBOOK

VIC-20 USER'S HANDBOOK

EPSON PRINTER USER'S HANDBOOK

The Epson FX, MX, and RX series printers are the most widely used printers for personal computers in the world. Epson printers have been hooked up to practically every kind of personal computer and used for every kind of printing job imaginable.

The EPSON PRINTER USER'S HANDBOOK takes the mystery out of hooking up and installing the Epson printer. It contains simple, explicit instructions and explanations so that even first-time users will be able to use their printers successfully. The book also contains a wealth of advanced information for more experienced users.

Topics covered in detail:

- How to hook up the printer and set the DIP and software switches that adapt the printer to different kinds of computers
- Using the Epson's impressive graphics capabilities
- How to take advantage of the Epson's ability to print italics, boldface, and other special characters and type fonts with popular word processing, spread sheet, and data base packages
- Tips on keeping the Epson running longer, faster, and more economically

The EPSON PRINTER USER'S HANDBOOK is a must for any owner or potential user of an Epson FX, MX, or RX series printer.

Computers/Machine Specific

31842
0 70999 00995
ISBN 0-345-31842-0

Cover printed in USA